

UNIVERSIDADE FEDERAL DO PARANÁ

3DBLAH

**Curitiba
2007**

ALEXANDRE PERIN MARCIANO DE SOUZA
FELIPE LATIF MECHAILECH
GABRIEL CARVALHO DA SILVA NEVES
RAFAEL SCUSSEL MICHELOTTO
RODRIGO CESAR DA SILVA

3DBLAH

**Trabalho de Conclusão de Curso apresentado à
Universidade Federal do Paraná, para obtenção do
título de Tecnólogo em Informática.**

Orientador: Professor Roberto Tadeu Raittz

Curitiba
2007

RESUMO

Este projeto orientado pelo professor Roberto Tadeu Raittz, tem por finalidade criar um *Instant Messenger*¹ um pouco diferente, no qual além de conversar e conhecer novas pessoas é possível selecionar um personagem (avatar) e interagir livremente em diversos ambientes em 3D. Para realizar qualquer ação dentro do sistema o usuário deve estar logado e conectado ao servidor principal. As ações referentes ao ambiente 3D são gerenciadas por um outro servidor menor, que também deve estar logado e conectado ao servidor principal.

¹ É um programa que além de enviar as mensagens, armazena os contatos do usuário, o perfil editado pelo mesmo, e configurações especiais do usuário, sendo possível até o envio de arquivos, desde que o remetente e o destinatário encontrem-se ambos conectados no servidor.

ABSTRACT

This project guided for the professor Roberto Tadeu Raittz, has for purpose to create an Instant Messenger a little different, where beyond talking and knowing new people it is possible to choose a character (an avatar) and to interact freely in diverse 3D environments. To carry through any action inside of the system the user must be log in and connected to the main server. The referring actions to the 3D environment are managed by one another lesser server, who also must be log in and connected to the main server.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	VII
RESUMO.....	VIII
ABSTRACT.....	IX
1 INTRODUÇÃO.....	01
2 ESCOPO E PROPÓSITO DO DOCUMENTO.....	02
2.1 PROPÓSITO DO DOCUMENTO.....	02
2.2 ESCOPO.....	02
2.3 APLICATIVOS AUXILIARES DO SISTEMA.....	02
3 OBJETIVOS DO PROJETO.....	04
3.1 OBJETIVOS GERAIS.....	04
3.2 OBJETIVOS ESPECÍFICOS DO 3DBLAH.....	04
3.3 DO SITE.....	05
3.4 DOS APLICATIVOS DOS SERVIDORES.....	05
4 JUSTIFICATIVA.....	06
4.1 DO 3DBLAH.....	06
4.2 DO SITE.....	06
4.3 DOS APLICATIVOS DOS SERVIDORES.....	06
5 CRONOGRAMA.....	08
5.1 FASES DO PROJETO.....	08
5.1.1. PLANEJAMENTO.....	08
5.1.2. DOCUMENTAÇÃO.....	08
5.1.3. DESENVOLVIMENTO.....	08
5.1.4. TESTES.....	09
6 FUNDAMENTAÇÃO TEÓRICA.....	10
6.1 DESCRIÇÃO.....	10
6.1.1 DESCRIÇÃO DO 3DBLAH.....	10
6.1.2 DO SITE 3DBLAH.....	12
6.1.3 DOS APLICATIVOS DOS SERVIDORES.....	13
6.2 METODOLOGIA.....	14

6.2.1 COREL DRAW X3.....	14
6.2.2 ADOBE PHOTOSHOP CS2.....	15
6.2.3 MACROMEDIA FLASH.....	15
6.2.4 MACROMEDIA DREAMWEAVER MX.....	16
6.2.5 PHP: HYPERTEXT PREPROCESSOR	17
6.2.6 DELPHI 7.....	18
6.2.7 TRUEVISION3D SDK 6.2.....	18
6.2.8 SQL SERVER 2000.....	18
6.2.9 VALVE HAMMER EDITOR 3.5.....	18
6.2.10 WALLY 1.55B.....	19
6.2.11 MILKSHAPE 3D 1.7.10.....	19
6.2.12 JUDE COMMUNITY 3.1.....	19
6.2.13 DB DESIGNER.....	19
6.2.14 MICROSOFT WINDOWS XP.....	19
6.2.15 MICROSOFT WORD 2003.....	20
6.3 CONCEITOS PESQUISADOS.....	20
6.3.1 AMBIENTE CLIENTE-SERVIDOR.....	20
6.3.2 SOBRE ENGINE.....	22
6.3.3 SOBRE MAPAS.....	22
6.3.4 INSTANT MESSENGER.....	22
7 DIAGRAMAS.....	23
7.1 DIAGRAMAS DE TELAS.....	23
7.1.1 TELAS DO SITE.....	23
7.1.1.1 TELA INICIAL DO SITE.....	23
7.1.1.2 TELA “CONHEÇA”	24
7.1.1.3 TELA DE CADASTRO.....	25
7.1.1.4 TELA DE ERRO NO CADASTRO.....	26
7.1.1.5 TELA DE CONFIRMAÇÃO DE CADASTRO.....	27
7.1.1.6 TELA DE DOWNLOAD.....	28
7.1.1.7 TELA DE ERRO NO LOGIN PARA DOWNLOAD.....	29
7.1.1.8 TELA DE MENU DE DOWNLOAD.....	30

7.1.1.9 TELA DE DOWNLOAD DE ATUALIZAÇÕES.....	31
7.1.1.10 TELA DE DOWNLOAD DE MAPAS.....	32
7.1.1.11 TELA DE DOWNLOAD DE MODELOS.....	33
7.1.1.12 TELA PÁGINA NÃO ENCONTRADA.....	34
7.1.2 TELAS DO APLICATIVO 3DBLAH.....	35
7.1.2.1 TELA DE LOGIN.....	35
7.1.2.2 TELA PRINCIPAL.....	37
7.1.2.3 TELA DE MAPA.....	38
7.1.2.4 TELA DE PERFIL (ABA DADOS PESSOAIS).....	40
7.1.2.5 TELA DE PERFIL (ABA PERSONAGEM).....	41
7.1.2.6 TELA DE CONVERSA.....	42
7.1.2.7 TELA DE PERFIL DE OUTRO USUÁRIO.....	43
7.1.2.8 TELA DE CONFIGURAÇÕES (ABA GERAL).....	44
7.1.2.9 TELA DE CONFIGURAÇÕES (ABA SONS).....	45
7.1.2.10 TELA DE CONFIGURAÇÕES (ABA DIRETÓRIOS).....	46
7.1.2.11 TELA DE CONFIGURAÇÕES (ABA CONTROLES).....	47
7.1.2.12 TELA DE CONFIGURAÇÕES (ABA MAPAS).....	48
7.1.3 TELAS DOS SERVIDORES.....	49
7.1.3.1 TELA DO SERVIDOR DE MAPAS.....	49
7.1.3.2 TELA PRINCIPAL DO SERVIDOR GERAL.....	50
7.1.3.3 TELA DE CADASTRO DE SERVIDOR DE MAPAS.....	51
7.1.3.4 TELA DE CADASTRO DE MODELOS.....	52
7.2 DIAGRAMA DE CLASSE.....	53
7.3 DIAGRAMA ER (FÍSICO).....	54
7.4 DIAGRAMAS DE CASO DE USO.....	55
7.4.1 DO APLICATIVO E SERVIDORES.....	55
7.4.2 DO SITE.....	57
7.5 DIAGRAMA DE ESTADOS DO SERVIDOR.....	57
7.6 DIAGRAMA DE ATIVIDADES.....	58
7.7 DESCRIÇÃO DE CASOS DE USOS, FLUXO DE EVENTOS E DIAGRAMA DE SEQUENCIA.....	60

7.7.1 CASOS DE USO DO 3DBLAH E DOS SERVIDORES.....	60
7.7.1.1 C.U. FAZER LOGIN.....	60
7.7.1.2 C.U. EDITAR PERFIL.....	61
7.7.1.3 C.U. EDITAR CONFIGURAÇÕES.....	63
7.7.1.4 C.U. ADICIONAR AMIGO.....	65
7.7.1.5 C.U. PESQUISAR AMIGO.....	67
7.7.1.6 C.U. ACEITAR AMIGO.....	68
7.7.1.7 C.U. APAGAR AMIGO.....	70
7.7.1.8 C.U. FAZER LOGOFF.....	72
7.7.1.9 C.U. CONECTAR À UM MAPA.....	73
7.7.1.10 C.U. ESTABELECEER CONVERSA.....	75
7.7.1.11 C.U. ENVIAR ARQUIVO.....	77
7.7.1.12 C.U. RECEBER ARQUIVO.....	79
7.7.1.13 C.U. MOVIMENTAR MODELO.....	81
7.7.1.14 C.U. CHAMAR ATENÇÃO.....	83
7.7.1.15 C.U. ATUALIZAR TELA.....	85
7.7.1.16 C.U. LIMPAR CONVERSA.....	86
7.7.1.17 C.U. SALVAR CONVERSA.....	88
7.7.1.18 C.U. DESCONECTAR USUÁRIO.....	89
7.7.1.19 C.U. DESCONECTAR TODOS.....	91
7.7.1.20 C.U. CADASTRAR MODELO.....	93
7.7.1.21 C.U. CADASTRAR SERVIDOR DE MAPAS.....	94
7.7.1.22 C.U. EDITAR SERVIDOR DE MAPAS.....	96
7.7.1.23 C.U. EDITAR MODELOS.....	98
7.7.1.24 C.U. EXCLUIR SERVIDOR DE MAPAS.....	100
7.7.1.25 C.U. EXCLUIR MODELOS.....	101
7.7.1.26 C.U. FAZER LOGIN.....	103
7.7.1.25 INTERPRETAR MENSAGENS.....	105
7.7.2 CASOS DE USO DO SITE.....	107
7.7.2.1 C.U. FAZER LOGIN.....	107
7.7.2.2 C.U. FAZER LOGOFF.....	108

7.7.2.3 C.U. ENVIAR MENSAGEM.....	110
7.7.2.4 C.U. DOWNLOAD.....	112
7.7.2.5 C.U. CADASTRAR SUÁRIO.....	114
8 CONCLUSÃO.....	122
9 REQUISITOS MÍNIMOS PARA O SISTEMA.....	123
9.1 PARA O SERVIDOR.....	124
9.2 PARA O CLIENTE.....	125
10 DIFICULDADES ENCONTRADAS.....	126
11 PROJETOS FUTUROS.....	127
12 AMBIENTE DE DESENVOLVIMENTO.....	122
13 BIBLIOGRAFIA.....	123

LISTA DE ILUSTRAÇÕES

Figura 1 - Tela inicial do Site.....	23
Figura 2 – Tela do site sobre o 3DBLAH.....	24
Figura 3 – Tela do site de cadastro.....	25
Figura 4 – Tela de mensagem de erro no cadastro.....	26
Figura 5 – Tela com mensagem de confirmação do cadastro.....	27
Figura 6 – Tela de login para download.....	28
Figura 7 – Tela com mensagem de erro do login.....	29
Figura 8 – Tela com os arquivos disponiveis para download.....	30
Figura 9 – Tela de mensagem para quando não existe atualizações.....	31
Figura 10 – Tela de mapas disponiveis para download.....	32
Figura 11 – Tela de modelos disponiveis para download.....	33
Figura 12 – Tela de página não encontrada.....	34
Figura 13 – Tela de login do 3DBLAH.....	35
Figura 14 – Tela principal do 3DBLAH.....	36
Figura 15 – Tela do mapa do sistema.....	37
Figura 16 – Tela do perfil do usuário do sistema.....	38
Figura 17 – Tela de escolha do modelo do usuário do sistema.....	39
Figura 18 – Tela de conversa entre os usuários.....	40
Figura 19 – Tela do perfil do outro usuário.....	41
Figura 20 – Tela de configuração do sistema em ambito geral.....	42
Figura 21 – Tela de configuração dos sons do sistema.....	43
Figura 22 – Tela de configuração de diretórios do sistema.....	44
Figura 23 – Tela de configuração dos controles do sistema.....	45
Figura 24 – Tela de configuração dos mapas do sistema.....	46
Figura 25 – Tela do servidor de mapa.....	47
Figura 26 – Tela do servidor principal.....	48
Figura 27 – Tela de cadastro de mapa no servidor de mapa.....	49
Figura 28 – Tela de cadastro de modelos.....	50
Figura 29 – Diagrama de Classes.....	57

Figura 30 – Diagrama de Entidade Relacionamento.....	58
Figura 31 – Diagrama de Caso de Uso do usuário.....	59
Figura 32 – Diagrama de Caso de Uso do servidos principal.....	60
Figura 33 – Diagrama de Caso de Uso do servidos de mapas.....	60
Figura 34 – Diagrama de Caso de Uso usuário em relação ao site.....	61
Figura 35 – Diagrama de Estado do Servidor.....	61
Figura 36 – Diagrama de Atividades.....	63
Figura 37 – Diagrama de Seqüência (Conectar ao servidor).....	65
Figura 38 – Diagrama de Seqüência (Fazer Login).....	67
Figura 39 – Diagrama de Seqüência (Editar Perfil).....	69
Figura 40 – Diagrama de Seqüência (Editar Configurações).....	71
Figura 41 – Diagrama de Seqüência (Adicionar Amigo).....	72
Figura 42 – Diagrama de Seqüência (Verificar Pedido).....	74
Figura 43 – Diagrama de Seqüência (Apagar Amigo).....	76
Figura 44 – Diagrama de Seqüência (Fazer Logoff).....	78
Figura 45 – Diagrama de Seqüência (Conectar ao Mapa).....	80
Figura 46 – Diagrama de Seqüência (Enviar Mensagem).....	82
Figura 47 – Diagrama de Seqüência (Enviar Arquivo).....	84
Figura 48 – Diagrama de Seqüência (Receber Arquivo).....	86
Figura 49 – Diagrama de Seqüência (Movimentar Modelo).....	87
Figura 50 – Diagrama de Seqüência (Pedir Atenção).....	89
Figura 51 – Diagrama de Seqüência (Atualizar Lista de Mapas).....	91
Figura 52 – Diagrama de Seqüência (Limpar Conversa).....	93
Figura 53 – Diagrama de Seqüência (Salvar Conversa).....	94
Figura 54 – Diagrama de Seqüência (Desconectar Usuário).....	96
Figura 55 – Diagrama de Seqüência (Desconectar Todos os Usuário).....	98
Figura 56 – Diagrama de Seqüência (Cadastrar Modelo).....	100
Figura 57 – Diagrama de Seqüência (Cadastrar Servidor de Mapas).....	102
Figura 58 – Diagrama de Seqüência (Editar Servidor de Mapas).....	103
Figura 59 – Diagrama de Seqüência (Editar Modelo).....	105
Figura 60 – Diagrama de Seqüência (Excluir Servidor de Mapa).....	107

Figura 61 – Diagrama de Seqüência (Excluir Modelo).....	109
Figura 62 – Diagrama de Seqüência (Fazer Login no Servidor de Mapa).....	111
Figura 63 – Diagrama de Seqüência (Interpretar Mensagem).....	112
Figura 64 – Diagrama de Seqüência (Fazer Login no Site).....	114
Figura 65 – Diagrama de Seqüência (Fazer Logoff do Usuário do Servidor)	116
Figura 66 – Diagrama de Seqüência (Enviar Mensagem via Site).....	118
Figura 67 – Diagrama de Seqüência (<i>Download</i> de Arquivos do Site).....	120
Figura 68 – Diagrama de Seqüência (Cadastro do Usuário no Site).....	122

1 INTRODUÇÃO

Este projeto foi desenvolvido devido à percepção da carência de ferramentas e aplicativos que permitam interação entre usuários e cenários reproduzidos virtualmente em um ambiente *online*, nos quais é possível o gerenciamento de conversas, trocas de arquivos e até mesmo uma forma de turismo virtual.

Percebemos uma possibilidade de criar algo inovador e criativo, que possam nos proporcionar muitos clientes e futuramente lucros também. Não será fácil, pois o ambiente cliente-servidor é um pouco complicado para manipular. Portanto o tema foi escolhido de acordo com as aptidões de cada integrante da equipe e com um consenso geral entre equipe e orientador.

Neste projeto de conclusão de curso buscamos desenvolver as principais funcionalidades do programa, como envio de mensagens, adição de contatos, edição de perfil e configurações, inserção de modelos em pelo menos dois ambientes virtuais, sendo eles uma sala teste e uma recriação virtual da Escola Técnica da Universidade Federal do Paraná.

2 ESCOPO E PROPÓSITO DO DOCUMENTO

2.1 Propósito do Documento

Este documento serve como base para o desenvolvimento do aplicativo 3DBLAH, no qual se encontrarão descritas todas as funcionalidades, objetivos e metodologias usadas no sistema. Este projeto está destinado ao Trabalho de Conclusão de Curso para a graduação em Tecnologia em Informática da Universidade Federal do Paraná sob orientação do professor Roberto Tadeu Raittz.

2.2 Escopo

Criar um Instant Messenger com um ambiente 3D, que tem como propósito uma comunicação em tempo real através de mensagens, enviadas ao contato de destino no mesmo momento em que fora solicitada. O sistema a ser desenvolvido disponibilizará a seus usuários além do serviço de conversa em tempo real, a possibilidade de cada contato possuir um personagem 3D (seu avatar), o qual poderá interagir com um cenário recriado tridimensionalmente, explorando-o e interagindo com outros usuários, conhecendo novas pessoas e novos locais. Além de viajar virtualmente ainda poderão manter contatos e amizades no programa, através de uma lista de contatos pessoal, onde poderão ser visualizados o perfil de seus amigos e enviar mensagens, arquivos e ainda poderão ser utilizadas outras funcionalidades comuns em *Instant Messengers*.

2.3 Aplicativos Auxiliares do Sistema

O software 3DBLAH necessita de alguns aplicativos auxiliares para funcionar perfeitamente, e para isso desenvolvemos os mesmos. Estes programas são: um site e dois aplicativos para o controle de servidores: um para o servidor de cenários e um para o servidor geral. O site foi desenvolvido para a divulgação e a alimentação do banco de dados do software, adequando-os aos mesmos padrões encontrados em sistemas

semelhantes. Oferecendo o *download* do produto, a descrição, o cadastro e informações especiais, bem como um suporte on-line.

Os aplicativos para administrar os servidores possuem basicamente a mesma funcionalidade, diferenciando-se da seguinte maneira: um gerencia as ações realizadas no ambiente tridimensional e o outro gerencia as funcionalidades comuns e próprias de um Instant Messenger. Foram desenvolvidos para que o servidor principal não ficasse sobrecarregado, dividindo as funcionalidades e possibilitando que o usuário mantenha acesso aos servidores de maneira independente.

3 OBJETIVOS DO PROJETO

3.1 Objetivos Gerais

O projeto tem por objetivo geral desenvolver um sistema que permita ao usuário conhecer lugares virtualmente, e ainda poder conversar com amigos e até mesmo conhecer pessoas novas, interagindo com elas tridimensionalmente. É um novo conceito de bate-papo. O Objetivo principal é a diversão e entretenimento do usuário, que pode aumentar sua lista de amigos e seu conhecimento, através do “turismo” e a história do cenário onde se encontra.

3.2 Objetivos Específicos do 3DBLAH

As funcionalidades deste sistema advêm da concepção de um Instant Messenger atual, como o MSN da Microsoft. O programa visa a interação do usuário com seus amigos em um cenário tridimensional, podendo ainda desfrutar das vantagens e funcionalidades ao utilizar o programa.

As funcionalidades presentes no programa, os objetivos específicos que esperamos alcançar, serão: a criação de um personagem, o qual o usuário movimentará pelo cenário; o usuário logar-se e sair do programa ou da conta; o usuário conectar e desconectar dos servidores; a adição de contatos à sua lista de amigos; possibilidade de edição do seu perfil e visualização do perfil de outros contatos; possibilidade de envio de arquivos e mensagens; personalização de status (ausente, *online*, *offline*, etc.); e um suporte no programa.

O desempenho da aplicação, em nível de velocidade de processamento de imagens, mensagens, requisições e demais dados, dependerá do hardware e da conexão de internet do computador onde esta será hospedada o servidor do programa. Por ser uma aplicação voltada para o esquema cliente-servidor, a velocidade de conexão e o hardware do usuário também implica em seu desempenho nos mesmos quesitos ao utilizar o programa, portanto pretendemos executá-las em máquinas de boas configurações para obter o melhor resultado possível.

3.3 Do Site

O desenvolvimento do site tem por objetivo disponibilizar informações sobre o funcionamento e sobre o aplicativo, além de disponibilizar o download do programa, junto com novos mapas, modelos e atualizações. O cadastro de cada usuário do programa pode ser feito apenas pela Internet e através da nossa *home page*, criando sua conta e login que serão usados para o download e para o login no 3DBLAH.

Outro objetivo que foi buscado é um suporte *online*, que pode ser encontrado na sessão Contato do site. Através dele receberemos mensagens com críticas, dúvidas e elogios, respondendo o mais breve possível.

3.4 Dos Aplicativos dos Servidores

Os aplicativos dos servidores têm como objetivos principais interligar e administrar as conexões dos servidores com clientes e entre os próprios servidores, possibilitando ainda que o administrador dos servidores realize ações com os clientes, mantendo uma ordem nos mapas e no devido uso do programa.

O objetivo do servidor principal é gerenciar as trocas de dados entre ele e o servidor de mapas e também entre ele e os usuários. Já o objetivo do servidor de mapas é administrar as trocas de dados entre os usuários conectados a ele.

4 JUSTIFICATIVA

4.1 Do 3DBLAH

O projeto foi desenvolvido com base na carência presente no mercado de programas que simulem uma realidade virtual tridimensional, e junto com ele surgiu a idéia de imbuir um Instant Messenger. Além da ambientação 3D, a exploração de cenários ou pontos turísticos virtuais e a interação de avatares nestes ambientes, o programa ainda conta com a maioria dos recursos presentes hoje em gerenciadores de mensagens instantâneas.

A idéia não é de toda originalidade, pois existe um Instant Messenger (IMVU) que visa entreter os usuários virtualmente, porém o programa é voltado para o público jovem, estando presentes muitos ambientes e avatares não realistas. O 3DBLAH surge com o intuito de entreter todas as faixas etárias possíveis, simulando de maneira mais real uma sala de bate papos, ou lugares onde haja possibilidade de fluxo de pessoas.

4.2 Do Site

Ao desenvolvermos o aplicativo 3DBLAH notamos a necessidade do mesmo possuir um site no qual fosse possível ter conhecimento do programa e suas funcionalidades antes de adquiri-lo, afinal é uma aplicação voltada à Internet, portanto não faria sentido não possuir sua própria home page. Atualmente, todos os aplicativos desenvolvidos possuem um suporte *online* em seus próprios sites e esta é a principal justificativa do desenvolvimento da página 3DBLAH. Além da divulgação do nosso produto.

4.3 Dos Aplicativos dos Servidores

Durante o desenvolvimento do aplicativo 3DBLAH, vimos a necessidade de um programa para gerenciar as trocas de dados entre cliente-servidor, cliente-servidor de mapas e servidor-servidor de mapas. Foram desenvolvidos dois servidores para não sobrecarregar o servidor principal do programa e deixar os serviços oferecidos indisponíveis.

5 CRONOGRAMA

Abaixo a descrição do cronograma, obtido com uso das técnicas de estimativas.

5.1 Fases do Projeto

5.1.1 Planejamento

a Levantamento de Requisitos

Data de Início: 15/03/06 Data de Término: 31/03/06

b Apresentação do Plano Geral de Projeto

Data de Início: 01/04/06 Data de Término: 30/04/06

5.1.2 Documentação

a Diagrama de Casos de Uso

Data de Início: 01/12/06 Data de Término: 15/01/07

b Diagrama de Classes

Data de Início: 08/05/06 Data de Término: 15/05/06

c Diagrama de Sequência

Data de Início: 07/12/06 Data de Término: 26/01/07

d Modelo Entidade-Relacionamento

Data de Início: 24/05/06 Data de Término: 31/05/06

5.1.3 Desenvolvimento

a Modelo lógico do banco de dados

Data de Início: 01/06/06 Data de Término: 15/06/06

b Módulo Administrativo Global

Data de Início: 16/06/06 Data de Término: 15/08/06

c Módulo Administrativo do Projeto

Data de Início: 16/08/06 Data de Término: 16/09/06

d Módulo Acompanhamento

Data de Início: 17/09/06 Data de Término: 31/10/06

e Programação do Aplicativo

Data de Início: 01/06/06 Data de Término: 12/02/07

f Programação do Site

Data de Início: 01/06/06 Data de Término: 02/01/07

5.1.4 Testes

a Testes Periódicos

Data de Início: 01/06/06 Data de Término: 28/02/07

b Testes Finais

Data de Início: 12/02/07 Data de Término: 28/02/07

6 FUNDAMENTAÇÃO TEÓRICA

6.1 Descrição

Neste tópico veremos as descrições dos aplicativos desenvolvidos para o 3DBLAH, como o site, o programa em si e os servidores.

6.1.1 Descrição do 3DBLAH

Para executar o 3DBLAH é necessário primeiramente adquiri-lo através da internet, após ter realizado seu cadastro no site, possuindo já seu login e sua senha. Através de um assistente *wizard* é realizada a instalação do aplicativo, caso tenham sido aceitado os termos de compromisso, a instalação é simples, com escolha de diretórios e criação de ícones de atalho para o 3DBLAH na sua área de trabalho.

Com o programa instalado para executá-lo é necessário o seu login e sua senha, cadastrados no site, a partir deste passo inicia-se o aplicativo. Primeiramente é validado o login. É aberta então a tela principal, com seus dados, sua lista de contatos, cenários disponíveis para conexão, opções para editar perfil, configurações, ajuda etc. A partir daí já é possível se comunicar com seus amigos da lista de contatos.

Quando algum contato pede autorização para fazer parte de sua lista de contatos aparece imediatamente uma janela, na qual você decide se aceita o usuário ou não. É possível ainda que você adicione um contato através da opção adicionar contato junto à lista de contatos, na qual você insere o e-mail do usuário a adicionar.

Sempre que algum contato da sua lista conecta-se ao 3DBLAH, uma janela de *pop-up* se abre no canto da tela avisando que o usuário conectou-se. É possível iniciar uma conversa com o contato apenas clicando nesta janela.

Para conectar-se a um cenário 3D basta clicar duas vezes no cenário escolhido ou clicar uma vez no botão “Conectar” no canto esquerdo da tela. Para atualizar esses cenários disponíveis para comunicação, deve-se clicar no botão “Atualizar”.

O botão “Perfil” abre a tela de configuração de perfil, na qual podem ser editados seus dados pessoais e seu avatar. Já o botão “Configurar” é para configuração do programa em si, como sons, diretórios, controles etc.

Caso o usuário encontre algum tipo de dificuldade para utilização do programa, basta selecionar a opção “Ajuda”.

Para adicionar um contato à sua lista basta clicar no ícone com um boneco e um “+” e seguir os passos descritos na janela que se abrir. Para excluir um contato basta clicar no ícone ao lado deste, com um boneco e um “-”, e seguir os passos da janela que se abrirá.

O usuário pode configurar seu status mais facilmente através da combobox no acima da sua lista de contatos, abaixo do rótulo “Status”, ainda no canto superior direito da tela.

Ao clicar no botão “Logoff” o usuário é desconectado do servidor 3DBLAH e é direcionada novamente a tela inicial de login do programa. Ao escolher a opção “Sair”, além de o usuário ser desconectado, o programa é finalizado.

Na janela de bate-papo é possível enviar mensagens através do campo de texto onde é digitada a mensagem e após um clique no botão “Enviar”. Pode limpar a conversa inteira apenas com um clique no ícone “Limpar Conversa” ou salvá-la através do ícone “Salvar”. É possível chamar a atenção de um contato através do botão “Chamar Atenção”, o qual envia um aviso sonoro e visual ao contato.

Ao conectar-se a um mapa novas opções apareceram no canto inferior da tela, como a opção *screenshot* que ao ser executada tira uma “foto” do mapa 3D em que você está, exatamente como você está o vendo. Esta imagem é salva diretamente na pasta de screenshots criada ao ser instalado o programa.

No ambiente 3D é possível explorar, caminhar e até encontrar amigos e novas pessoas, podendo iniciar uma conversa com essa pessoa caso ela esteja a um raio mínimo de distância do avatar do usuário. É possível visualizar no canto esquerdo inferior da tela do cenário o *nick* do usuário que está com o avatar próximo do modelo do usuário.

Resumindo, o funcionamento do sistema é basicamente assim: o usuário pede conexão ao servidor, servidor valida o login e autoriza conexão, então o servidor

disponibiliza a lista de contatos e os mapas para o sistema, para ser mostrado ao usuário. As trocas de mensagens e arquivos são transferidas ao servidor e este transfere ao cliente de destino.

6.1.2 Do Site 3DBLAH

O site dispõe de uma tela principal de um menu com as opções Início, Conheça, Cadastro, Download e Contato.

A opção Início remete o usuário à página principal do site, onde haverá um edital sobre novas atualizações e downloads, com algumas imagens do aplicativo.

Na opção Conheça, está disponível uma descrição detalhada do aplicativo e suas funcionalidades, com imagens e textos explicativos.

Já a opção de Cadastro permite ao usuário que deseja fazer parte do mundo 3DBLAH, que crie sua conta, para que possa realizar o download do aplicativo e então instalá-lo em sua máquina. Nessa tela são preenchidos seus dados pessoais.

Na seção Download são disponibilizados as novas, atualizações, modelos, mapas e o programa em si, para ser adquirido. É necessário realizar o login para ter acesso à essa página. Depois de validado o usuário e senha, abre-se a nova tela com as seções e quantidades de arquivos disponibilizados em cada seção. Então são visualizados todos arquivos existentes para download conforme a seção escolhida.

Ainda existe a possibilidade de entrar em contato conosco, através da opção Contato. Nessa tela existem caixas de texto para Identificação do usuário e para a mensagem a ser enviada, sendo ela uma dúvida, reclamação ou elogio.

Quando o usuário é cadastrado, o site envia os dados diretamente ao banco de dados, que é comum a todas as aplicações que englobam o 3DBLAH.

6.1.3 Dos Aplicativos dos Servidores

O desenvolvimento de um servidor para a aplicação 3DBLAH é vital para o programa, pois através dele é permitida a troca de dados entre os usuários. Como o aplicativo utiliza-se de uma ambientação gráfica tridimensional, vimos a necessidade de

dividir o servidor em dois, um que manipule essa conexão e trocas de informações entre os usuários dentro dos mapas e outro que gerencie as funcionalidades normais de um Instant Messenger, este ainda possui conexão com o primeiro, para agilizar e facilitar a administração do sistema.

O servidor principal possui um aplicativo, que torna possível a administração dos usuários conectados ao programa, podendo desconectá-los a qualquer momento, bloquear o uso do programa para os mesmos, e ainda permite cadastrar, alterar e excluir os mapas e modelos no banco de dados. Este servidor que executa as trocas de mensagens, arquivos, edição de perfis, adição de contatos, alteração de status, enfim, todas as tarefas que envolvem o Banco de Dados e que não possuem uma manipulação 3D.

A tela de cadastro de avatares (modelos), possui uma caixa do lado direito, na qual pode ser visualizado o modelo a ser adicionado, o nome que receberá e suas características. Também é possível consultar, editar e excluir o modelo na base de dados, porém os arquivos dos modelos devem estar presentes nas máquinas dos usuários, para que possam ser utilizados no programa.

Existe ainda no servidor principal a possibilidade de cadastrar, consultar, editar e excluir os mapas a serem utilizados para o programa. Nesta interface, está disposta uma caixa para que seja adicionada uma foto para o mapa, que será visualizada na tela principal do 3DBLAH.

Os avatares e os mapas possuem um campo chamado link, o qual será disposto no site, na sessão de download, permitindo uma alimentação automática dos downloads do site.

O segundo servidor é referente aos mapas disponíveis para o programa e às ações realizadas dentro desse ambiente 3D, como andar e interagir com outros usuários. O aplicativo deste, é muito simples e apenas mostra as características do mapa disponibilizado e uma lista de usuários conectados a eles, com seus respectivos *ips* e *emails* listados.

Os servidores necessitam de portas específicas disponíveis para rodar. São elas as portas: 2255 para o servidor principal e a 2256 para o servidor de mapas. Da mesma

maneira, o cliente também deve possuir as portas especificadas disponíveis, para estabelecer comunicação com os servidores.

6.2 Metodologia

A metodologia refere-se aos programas e técnicas usados para o desenvolvimento do projeto, portanto a seguir será comentado sobre todos softwares utilizados para o desenvolvimento dos aplicativos.

6.2.1 Corel Draw X3

Para o design do site foi usado o CorelDraw X3, que por ser uma ferramenta muito versátil e que fornece recursos para desenvolver qualquer layout. Um ponto negativo dessa ferramenta é a exportação de imagens, pois ao serem exportadas apresentam bordas bancas em volta delas. O CorelDRAW surgiu em [1989](#), apenas em [inglês](#). Em [1995](#), surgiu a primeira versão em [32 bits](#) (CorelDRAW 6). Dois anos depois surgiu a primeira versão para computadores [Macintosh](#). No ano seguinte, foi lançada a primeira versão para [Linux](#). Em [2003](#), surgiu a versão 12 que está preparada para o [Windows XP](#). Sendo que a última versão criada no início de 2006 se denomina corel X 3 ("X" em algarismos romanos = 10 + 3 = versão 13). Tem como concorrentes directos principais, os programas [Adobe Illustrator](#) e [Macromedia Freehand](#), em ambiente [software proprietário](#), já em [software livre](#) o [Inkscape](#). (<http://pt.wikipedia.org/wiki/CorelDRAW>, acessado em 16 de janeiro de 2007.)

6.2.2 Adobe Photoshop CS2

Usamos o Adobe Photoshop CS2 para o tratamento de imagens principalmente, sendo utilizado para tratar imperfeições das imagens exportadas pelo CorelDraw X3. Adobe Photoshop é um editor de imagens bidimensionais do tipo [raster](#) (possuindo ainda algumas capacidades de edição típicas dos editores [vetoriais](#)) desenvolvido pela [Adobe Systems](#). É considerado o líder no mercado dos editores de imagem profissionais, assim como o programa *de fato* para edição profissional de imagens

digitais e trabalhos de [pré-impressão](#). A versão utilizada *Adobe Photoshop CS2* (sigla cujo significado é "creative suite 2", correspondente à nona edição desde seu lançamento), é disponível para os sistemas operacionais [Microsoft Windows](#) e [Mac OS X](#). As versões anteriores até à 7.0 podem ser usadas em outros sistemas operativos, como o [Linux](#), por meio de programas como o [CrossOver Office](#). Algumas versões anteriores foram lançadas também para [IRIX](#), mas o suporte a esta versão foi descontinuado após a versão 3.0. (<http://pt.wikipedia.org/wiki/Photoshop>, acessado em 16 de janeiro de 2007.)

6.2.3 Macromedia Flash

O Macromedia Flash foi utilizado para fazer a animação de todo o cabeçalho do site, tornando mais interativo para os usuários. Adobe Flash (antigamente Macromedia Flash), ou simplesmente Flash, é um programa gráfico vetorial utilizado para se criar [animações](#) interativas, desenvolvido e comercializado pela [Macromedia](#) (empresa especializada em desenvolver [programas](#) que auxiliam o processo de criação de páginas web). Em abril de 2005 a Macromedia foi adquirida pela Adobe System, fabricante de diversos programas gráficos conhecidos como, por exemplo, o [Photoshop](#).

Os [arquivos](#) gerados pelo Flash, chamados de "SWF" (Shockwave Flash File), podem ser visualizados em uma página web usando um navegador web, ou utilizando-se o Flash Player que é um pequeno aplicativo distribuído gratuitamente pela Adobe. Os arquivos feitos em Flash são comumente utilizados para propaganda animada (*banners*) em páginas web, mas não limitando-se a isso, pois há também em abundância vários jogos e apresentações dos mais variados tipos utilizando tecnologia Flash, na [Internet](#). Em versões recentes (a partir da 5), a Macromedia expandiu a utilização do Flash para além de simples animações, mas também para uma ferramenta de desenvolvimento de aplicações completas. Isso graças aos avanços na linguagem [ActionScript](#) que é a linguagem de programação utilizada em aplicações SWF. A terceira versão desta linguagem acaba de ser lançada tornando mais fácil e rápido criar aplicações para web mas, no entanto, a segunda versão ainda é a mais utilizada no mercado. Uma nova plataforma, chamada [Apolo](#) está sendo lançada pela Adobe e tem por objetivo solidificar o desenvolvimento da

linguagem ActionScript, seja através do Flash, do [Macromedia Flex](#) ou de outros programas. (http://pt.wikipedia.org/wiki/Macromedia_Flash, acessado em 16 de janeiro de 2007)

6.2.4 Macromedia Dreamweaver MX

O Macromedia Dreamweaver MX foi o aplicativo utilizado para a programação e diagramação do site, tendo como linguagem usada, o PHP5, por ser dinâmica e gratuita. O Macromedia Dreamweaver é uma ferramenta de desenvolvimento para a [web](#) criada pela [Macromedia](#) (agora [Adobe Systems](#)), que está atualmente na [versão](#) 8. Versões iniciais da aplicação serviam como um simples editor [HTML WYSIWYG](#) ("What You See Is What You Get", ou "O que você vê é o que você recebe"), porém versões mais recentes incorporaram notável suporte para muitas outras tecnologias web tais como [XHTML](#), [CSS](#), [JavaScript](#) e vários scripts de servidor. Desde o final dos anos 90, o Dreamweaver vem tendo um sucesso crescente e hoje domina cerca de 80% do mercado de editores HTML. Existem versões tanto para [Mac OS](#) quanto para [Windows](#), mas também é possível executá-lo em plataformas [Unix](#) e sabores através do uso de softwares de emulação como o Wine.

Com o advento da versão MX, a Macromedia incorporou ferramentas de criação de conteúdo dinâmico ao Dreamweaver. No espírito das ferramentas HTML WYSIWYG, permite que usuários se conectem a bancos de dados (tais como [MySQL](#) e Microsoft [Access](#)) para filtrar e mostrar conteúdo usando tecnologias de script tais como [PHP](#), [ColdFusion](#), [ASP](#) e [ASP.NET](#), sem qualquer experiência prévia em programação.

6.2.7 PHP: Hypertext Preprocessor

PHP (um [acrônimo](#) recursivo para "PHP: Hypertext Preprocessor") é uma [linguagem de programação](#) de computadores interpretada, [livre](#) e muito utilizada para gerar conteúdo dinâmico na [Web](#). Apesar de ser uma linguagem de fácil aprendizado e de uso para pequenos scripts dinâmicos simples, o PHP é uma linguagem poderosa orientada a objetos.

A linguagem surgiu por volta de [1994](#), como um subconjunto de scripts [Perl](#) criados por [Rasmus Lerdorf](#), com o nome PHP/FI (Personal Home Page Tools/Forms Interpreter). Com as adições de [Zeev Suraski](#) e [Andi Gutmans](#),

dois programadores [israelitas](#) pertencentes ao [Technion](#), o Instituto Israelita de Tecnologia, que reescreveram o *parser*, era lançada em [1997](#) a PHP 3, primeira versão estável e parecida com a linguagem actual. Ao reescrever o *parser*, foi criado o [Zend Engine](#), que é mantido oficialmente pela empresa [Zend](#) em conjunto com a comunidade PHP. Em Maio de [2000](#) veio a público a versão 4, e em Julho de [2004](#), a versão 5, onde a principal mudança foi uma nova [API](#) para [orientação a objetos](#) provida pelo [Zend Engine](#) 2.

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores *web*. Diversos módulos são criados no repositório de extensões [PECL](#) (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem [C](#) e com a [C++](#). Pode ser, dependendo da configuração do servidor, embutida no código [HTML](#). Além disso, destaca-se a extrema facilidade com que PHP lida com servidores de [base de dados](#), como [MySQL](#), [Firebird](#), [PostgreSQL](#), [Microsoft SQL Server](#) e [Oracle](#).

Existem versões do PHP disponíveis para os seguintes sistemas operacionais: [Windows](#), [Linux](#), [FreeBSD](#), [Mac OS](#), [OS/2](#), [AS/400](#), [Novell Netware](#), [RISC OS](#), [IRIX](#) e [Solaris](#)

A Wikipédia funciona sobre um software inteiramente escrito em PHP, usando bases de dados [MySQL](#): o [MediaWiki](#).

Construir uma página dinâmica baseada em bases de dados é simples com PHP, este provê suporte a um grande número de bases de dados: [Oracle](#), [Sybase](#), [PostgreSQL](#), [InterBase](#), [MySQL](#), [SQLite](#), [MSSQL](#) etc, podendo abstrair o banco com a biblioteca [ADOdb](#), entre outras.

PHP tem suporte aos protocolos: [IMAP](#), [SNMP](#), [NNTP](#), [POP3](#), [HTTP](#), [LDAP](#), [XML-RPC](#), [SOAP](#). É possível abrir [sockets](#) e interagir com outros [protocolos](#). E as bibliotecas de terceiros expandem ainda mais estas funcionalidades.

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos. A mais notável é a [PHP-GTK](#). Trata-se de um conjunto do PHP com a biblioteca [GTK](#), portada do [C++](#), fazendo assim softwares inter-operacionais entre [Windows](#) e [Linux](#).

(<http://pt.wikipedia.org/wiki/Php> , acessado em 16 de janeiro de 2007)

Foi usado para o desenvolvimento do site 3DBLAH, em sua versão 5, orientada a objetos.

6.2.6 Delphi 7

O Delphi é uma linguagem de programação muito conhecida e utilizada nos dias de hoje, mesmo tendo que pagar a sua licença de uso. Ele é orientado a objetos, com um forte potencial gráfico para interface. Foi escolhido para desenvolver o projeto devido a um maior conhecimento e facilidade de entendimento e manipulação de todos integrantes da equipe para com a linguagem, além de apenas ele oferecer a compatibilidade com os programas a serem utilizados, os gráficos e um bom manipulamento dos dados com o banco de dados escolhido.

6.2.7 TrueVision3D SDK 6.2

A ferramenta gráfica TrueVision3D foi desenvolvida por Sylvian Dupont em janeiro de 1999 e começou apenas como um hobby, porém com o passar do tempo foi ganhando força com novos agregados ao projeto, tendo sua primeira versão para o mercado em 2002. Ela foi utilizada em nosso programa para toda a ambientação 3D, para a manipulação dos mapas e dos modelos também.

6.2.8 SQL Server 2000

O SQL Server 2000 é muito conhecido pela sua velocidade e praticidade de uso, e devido a isso foi escolhido para ser o banco de dados do projeto, o qual precisa de rápidas transferências de dados devido às conexões cliente-servidor constantes no escopo do projeto.

6.2.9 Valve Hammer Editor 3.7

O Valve Hammer Editor 3.7 é uma atualização do Valve Hammer Editor 3.4. Ele é um editor de mapas tridimensionais, ou seja, a partir deste programa pudemos desenvolver os cenários localizados no 3DBLAH. Os mapas criados e editados são compilados para o formato BSP, que é compatível com a engine TrueVision3D.

6.2.10 Wally 1.75B

O Wally é um programa gráfico para edição de arquivos WAD. Os arquivos WAD são como ZIP, eles contêm as texturas que serão utilizadas pelos mapas compactadas dentro deles.

6.2.11 MilkShape 3D 1.7.10

O MilkShape 3D foi usado para a modelagem e compilação dos personagens desenvolvidos para o projeto para o formato MDL, compatível com a engine TrueVision3D.

6.2.12 JUDE Community 3.1

O JUDE Community 3.1 é um programa utilizado para modelagem de dados, e esta é sua versão gratuita. Ele foi utilizado para o desenvolvimento dos diagramas de caso de usos, dos diagramas de seqüência e dos diagramas de estado.

6.2.13 DB Designer

É uma ferramenta gráfica para modelagem de banco de dados. Foi utilizado para auxiliar o desenvolvimento do diagrama de classes presente no projeto.

6.2.14 Microsoft Windows XP

É o sistema operacional que foi utilizado para o desenvolvimento do projeto estando presente em todos computadores da equipe, ou seja, é uma aplicação voltada para o Windows.

6.2.15 Microsoft Word 2003

É o editor de texto usado para o desenvolvimento da documentação do projeto, em específico da monografia a ser entregue. Ele é pertencente ao Microsoft Windows.

6.3 Conceitos Pesquisados

6.3.1 Ambiente Cliente-Servidor

Para um bom entendimento sobre o funcionamento do programa, vimos à necessidade de falar um pouco a respeito do ambiente cliente-servidor, com sua descrição, suas vantagens e desvantagens.

A filosofia cliente/servidor nasceu do conceito de processamento distribuído, originalmente idealizados pelos desenvolvedores de sistemas operacionais de rede. É aquele tipo de invenção que começa direcionada para aplicações complexas, e um dia acaba vindo parar em nossas mãos, como a suspensão ativa e o telefone celular.

Mas a idéia é muito simples. Você tem que envelopar 1.000 cartas, depois colá-las e pedir a um ajudante que as leve até o correio. Ora, que tal pedir para o ajudante ir colando algumas enquanto você ainda está envelopando outras? Isto é processamento distribuído, ou serviço distribuído. E é sobre este tão simples conceito que repousa a filosofia da programação cliente/servidor.

Desta forma, a máquina cliente (a máquina do usuário final) não fica encarregada de todo o processamento. Ela funciona como front-end, sendo utilizada basicamente para entrada e visualização de dados. O processamento mais pesado é feito no servidor.

O servidor recebe pedidos, ou requisições, do cliente e as processa, enviando imediatamente uma resposta.

Na prática, é necessário haver uma aplicação servidora, instalada em uma máquina, e uma aplicação cliente, instalada em outras máquinas. As aplicações cliente fazem pedidos à aplicação servidora, que as atende. Isto já deixa claro que o equipamento onde será instalada a aplicação servidora deve ser provido de maiores quantidades de recursos, tais como memória, disco e velocidade de processamento.

No mundo comercial de hoje, o principal objetivo das organizações é processar dados, e disparar processamentos a estes relacionados. Assim, a aplicação servidora é na verdade um servidor de banco de dados. Nele são concentrados todos os dados da empresa, e podem ser concentrados ainda os processamentos ligados a estes dados, sobre o que falaremos a frente.

(CAMPOS, André. "573 - Desvendando a programação "Client/Server".
Arquivo de ajuda do Borland Delphi 7)

O cliente, portanto, é qualquer sistema que acesse o servidor, no nosso caso é qualquer usuário ou servidor de mapas que logue-se no 3DBLAH.

Os programas que utilizam o ambiente cliente-servidor trafegam uma quantidade muito menor de dados pela rede. Isto se dá porque o cliente faz um pedido ao servidor, este pedido é então todo processado pelo servidor, retornando apenas uma pequena quantidade de dados para o cliente como resposta. Assim, as únicas transações em rede são o pedido do cliente e a resposta do servidor. O que torna a rede muito mais leve, com menos dados trafegando, o que resulta em um ganho na velocidade de processamento das aplicações.

E devido a isso foi escolhida uma aplicação neste ambiente, pois nem sempre o usuário possuiu uma máquina relativamente boa para que as tarefas sejam executadas localmente. O que torna recomendável o acesso à Internet banda larga para execução do 3DBLAH.

A segurança foi outro ponto que nos chamou muita atenção, pois é uma das mais notáveis vantagens deste ambiente, afinal apenas o servidor tem acesso ao banco de dados.

Pelos motivos citados acima vimos à importância e necessidade da adoção deste ambiente para a realização e execução do aplicativo 3DBLAH, facilitando todas as trocas de informações e velocidade de transferência de dados.

6.3.2 Sobre Engine

Engine: módulo pronto para ser utilizado (3D, Física, IA, Rede, etc).

Algumas empresas que desenvolvem jogos optam por desenvolver sua própria engine. Na verdade, existem vantagens e desvantagens ao se utilizar tecnologia desenvolvida por terceiros:

Vantagens: Rápido acesso à tecnologia; Economia de tempo e dinheiro; Bugs; Portabilidade; Atualizações.

Desvantagens: Bugs; Portabilidade; Descontinuidade; Prioridades.
(AZEVEDO, 2005, p. 115-116)

6.3.3 Sobre Mapas

Quando falamos em modelagem de cenários, temos a impressão que teremos que fazer objetos extremamente complexos. Na realidade, a imitação do número de polígonos em um game faz com que isto fique ainda mais difícil, pois você realmente terá de fazer um trabalho complexo, mas com recursos simples.

Um cenário completo, feito de forma fotorrealística, pode ter mais de 1 milhão de polígonos, enquanto um cenário para um game não pode ultrapassar 20 mil polígonos para ser jogável de fato. Por isto, é importante modelar o cenário dentro de uma ferramenta própria para este fim.

Muito do trabalho que deixa um cenário bem-feito vem da textura e iluminação, que veremos mais adiante. Você notará que muitas das janelas, portas e outros objetos são meramente ilustrativos, colocados pelas texturas, evitando que o engine precise compilar polígonos desnecessários.
(AZEVEDO, 2005, p. 117-118)

6.3.4 Instant Messenger

Ele é um programa que além de enviar as mensagens, armazena os contatos do usuário, o perfil editado pelo mesmo, e configurações especiais do usuário, sendo possível até o envio de arquivos, desde que o remetente e o destinatário encontrem-se ambos conectados e logados no servidor.

7 DIAGRAMAS

7.1 Diagramas de Telas

7.1.1 Telas do Site

7.1.1.1 Tela Inicial do Site



Figura 1: Tela inicial do Site.

Descrição: Esta é a tela inicial do site, a qual mostra algumas imagens do programa, juntamente com uma área de texto na qual são disponibilizadas algumas informações e novidades sobre o programa, ou seja, um edital.

7.1.1.2 Tela “Conheça”



Figura 2 – Tela do site sobre o 3DBLAH .

Descrição: Nesta tela é possível saber mais sobre o aplicativo 3DBLAH, conhecendo a história, a proposta e os requisitos mínimos para executar o programa.

7.1.1.3 Tela de Cadastro

Quarta-feira
14/02

Inicial | Conheça | Cadastro | Download | Contato

Cadastro 3D BLAH

Dados pessoais *campo obrigatório

Nome completo*:

Data nascimento*: / / dd/mm/aaaa

Sexo: ☐ masculino ☐ feminino

Estado civil:

Dados 3D BLAH

Nick (apelido)*: letras minúsculas, ex: joaopaulo

Usuário (e-mail)*: ex: nome@dominio.com.br

Senha*: máximo de 8 caracteres

Confirmar Senha*: inserir a mesma senha acima para fazer a confirmação

Dados adicionais

Cidade*:

Estado*:

País*: Brasil

Telefone: ex: (22)2222-2222

Celular: ex: (22)9999-9999

Figura 3 – Tela do site de cadatro.

Descrição: Esta tela é referente ao cadastro do usuário que deseja participar do 3DBLAH, onde existem campos obrigatórios para que o cadastro seja efetivado. Após os mesmos serem preenchidos basta clicar em Cadastrar para registrar sua conta.

7.1.1.4 Tela de Erro no Cadastro



Figura 4 – Tela de mensagem de erro no cadastro.

Descrição: Essa tela ocorre quando algum campo obrigatório não é preenchido na tela do cadastro ou quando há algum problema de conexão no momento do cadastro. Então há um link para voltar à página inicial do site.

7.1.1.7 Tela de Confirmação de Cadastro



Figura 5 – Tela com mensagem de confirmação do cadastro.

Descrição: Esta tela refere-se ao êxito em cadastrar um usuário, disponibilizando o link para que já possa ser realizado o download do programa.

7.1.1.6 Tela de Download



The screenshot shows a web browser window with the 3D BLAH logo in the top left corner. The date "Quarta-feira 14/02" is displayed in the top right. A yellow navigation bar contains the links: "Inicial", "Conheça", "Cadastro", "Download", and "Contato". The main content area is divided into two columns. The left column, titled "Já sou 3D BLAH", contains instructions for logging in and a form with fields for "usuário:" and "senha:", followed by an "ok" button. The right column, titled "Quero ser 3D BLAH", contains instructions for registration and a link that says "Quero me cadastrar agora mesmo!".

Quarta-feira
14/02

Inicial | Conheça | Cadastro | Download | Contato

Já sou 3D BLAH

Agora para adquirir o seu 3D BLAH basta você entrar com o seu **usuário** e **senha** que registrou em seu cadastro.

usuário:

senha:

Quero ser 3D BLAH

Para você se tornar uma 3D BLAH basta você fazer o seu cadastro. Depois baixe o programa para instalar em computador.

[Quero me cadastrar agora mesmo!](#)

Figura 6 – Tela de login para download.

Descrição: Esta tela possui na parte esquerda os campos necessários e obrigatórios para logar-se e realizar downloads referentes ao programa, como modelos e mapas. E no canto direito existe um link que remete o usuário à página de cadastro no caso de ele não ser cadastrado no sistema.

7.1.1.7 Tela de Erro no login para Download

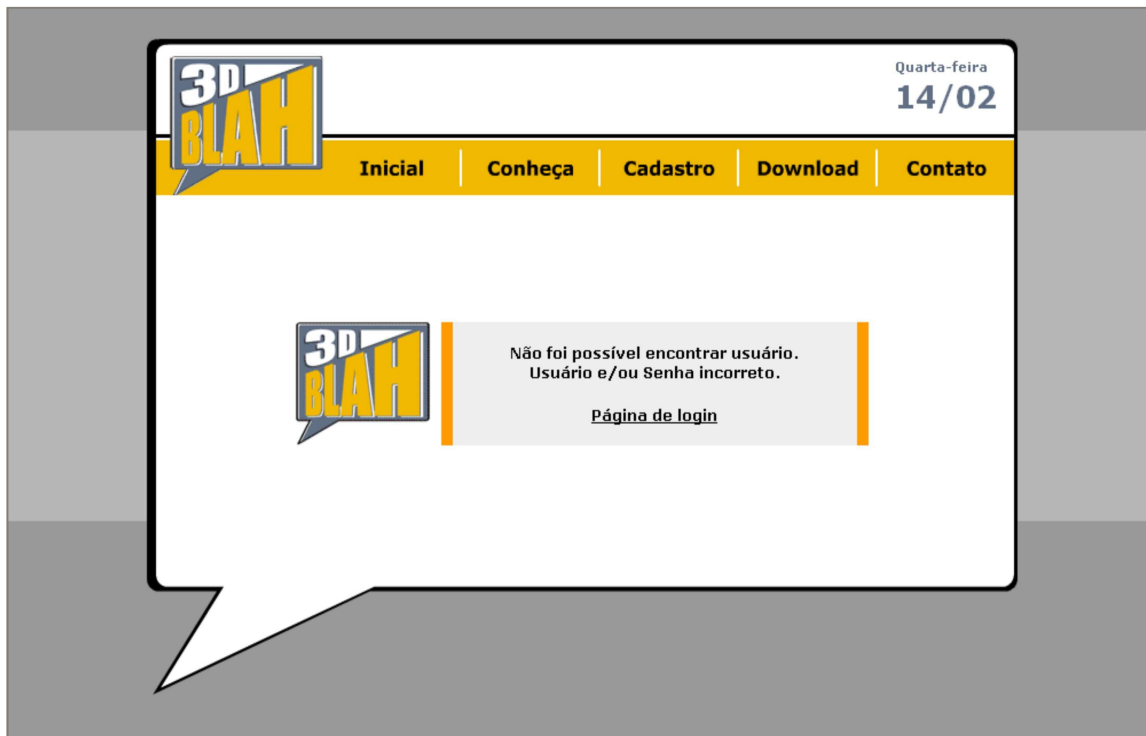


Figura 7 – Tela com mensagem de erro do login.

Descrição: Esta tela aparece quando o usuário comete um erro no login, e então é disponibilizado ao usuário o link para voltar à tela de login.

7.1.1.8 Tela de Menu de Download



Figura 8 – Tela com os arquivos disponíveis para download.

Descrição: Esta tela vem em sequência do login correto na página de downloads, nela estão dispostas as categorias disponibilizadas para downloads, no caso, o programa, os modelos, as atualizações e os mapas.

7.1.1.9 Tela de Download de Atualizações



Figura 9 – Tela de mensagem para quando não existe atualizações.

Descrição: Após escolher a opção “Atualizações” presente no item 6.1.1.8, o usuário é direcionado para esta página. Nesta tela estarão dispostas as atualizações referentes ao programa 3DBLAH.

7.1.1.10 Tela de Download de Mapas

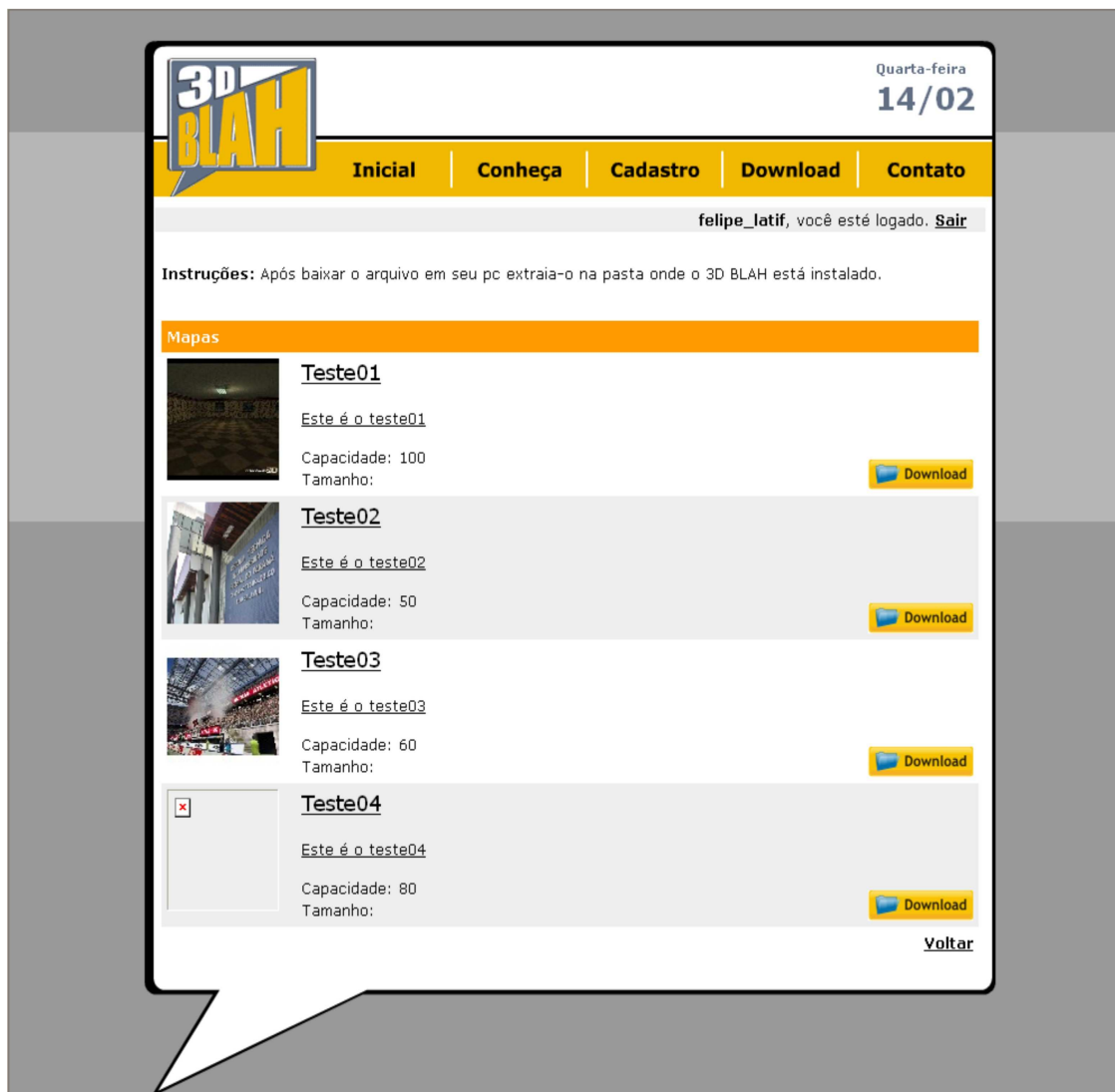


Figura 10 – Tela de mapas disponíveis para download.

Descrição: Após escolher a opção “Mapas” presente no item 6.1.1.8, o usuário é direcionado a esta página, a qual possui os mapas disponíveis com suas fotos, descrições, nomes, capacidade, o tamanho do arquivo e o link para o download. O link “Voltar” remete o usuário para a tela de menu de download.

7.1.1.11 Tela de Download de Modelos

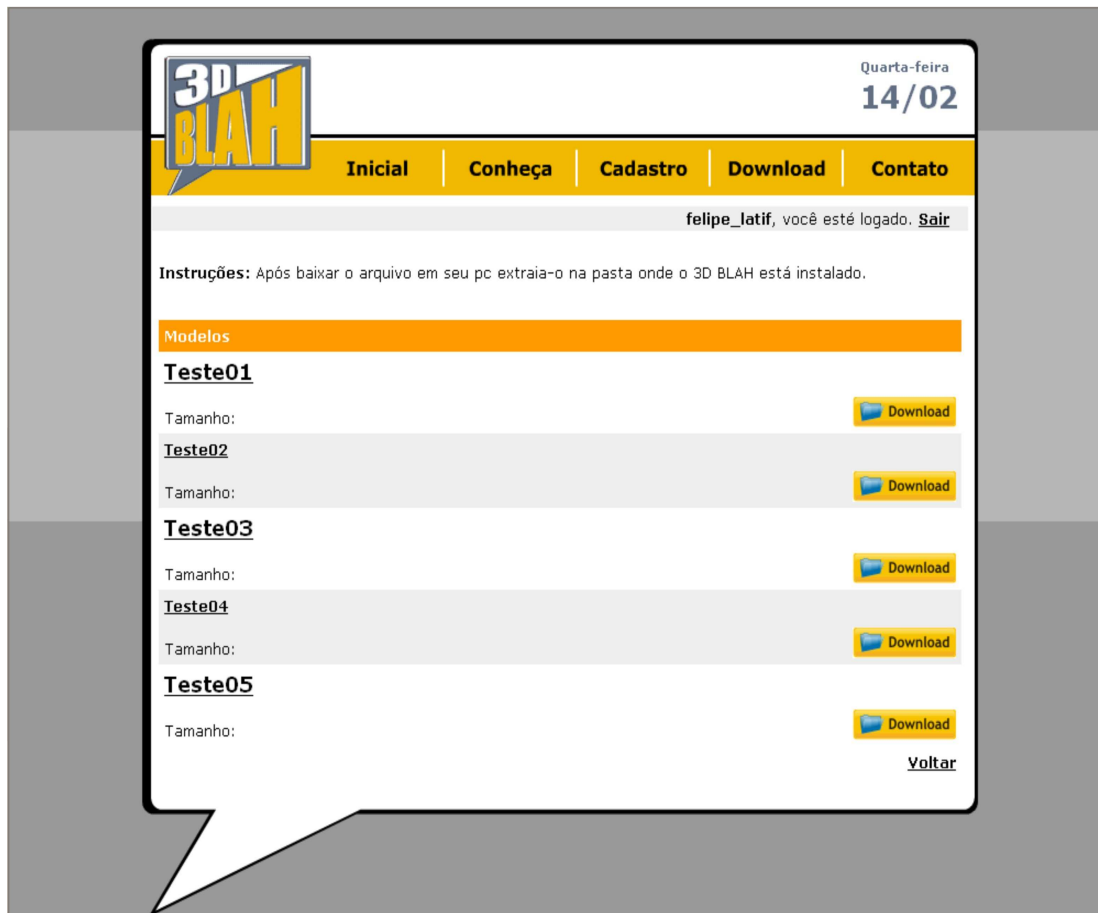


Figura 11 – Tela de modelos disponíveis para download.

Descrição: Após escolher a opção “Modelos” presente no item 6.1.1.8, o usuário é direcionado a esta página, a qual exibe uma relação dos modelos disponíveis, juntamente com seus respectivos tamanhos e links para downloads. O link “Voltar” remete o usuário a tela de menu de download.

7.1.1.12 Tela Página Não Encontrada



Figura 12 – Tela de página não encontrada.

Descrição: Esta aparece ao usuário sempre que houver um problema na sua conexão com o servidor, então é disposto um link que remeterá o usuário à página inicial do site.

7.1.2 Telas do Aplicativo 3DBLAH

7.1.2.1 Tela de Login



Figura 13 – Tela de login do 3DBLAH.

Descrição: Esta é a tela de login que aparece ao usuário assim que ele executa o programa. Nela existe um edital do que há de novo e está disponível no site para download. Abaixo estão presentes os campos para o login, com a possibilidade de

recuperar uma senha perdida ou mesmo criar uma nova conta a partir do botão “Novo Usuário”. É possível fechar o programa clicando em “Sair”.

7.1.2.2 Tela Principal



Figura 14 – Tela principal do 3DBLAH.

Descrição: Esta é a tela subsequente da tela de login, é a tela principal do programa. Nela estão dispostos os mapas disponíveis para conexão, a lista de contatos ordenada alfabeticamente, o apelido, o status e a imagem de exibição do usuário, juntamente com os botões Conectar-se (que conecta o usuário ao mapa escolhido), Atualizar (atualiza a lista de mapas disponíveis para conexão), Perfil (no qual o usuário pode configurar o seu perfil), Configurações (no qual o usuário edita algumas configurações básicas do programa), Ajuda (que abre o Help do programa), Log Off (no qual o usuário efetua o logoff do sistema e volta para a tela de login), Sair (que encerra o programa), Adicionar Contato (no qual o usuário adiciona um amigo a sua lista de contatos) e Apagar Contato (no qual o usuário exclui o contato selecionado de sua lista de contatos).

7.1.2.3 Tela de Mapa

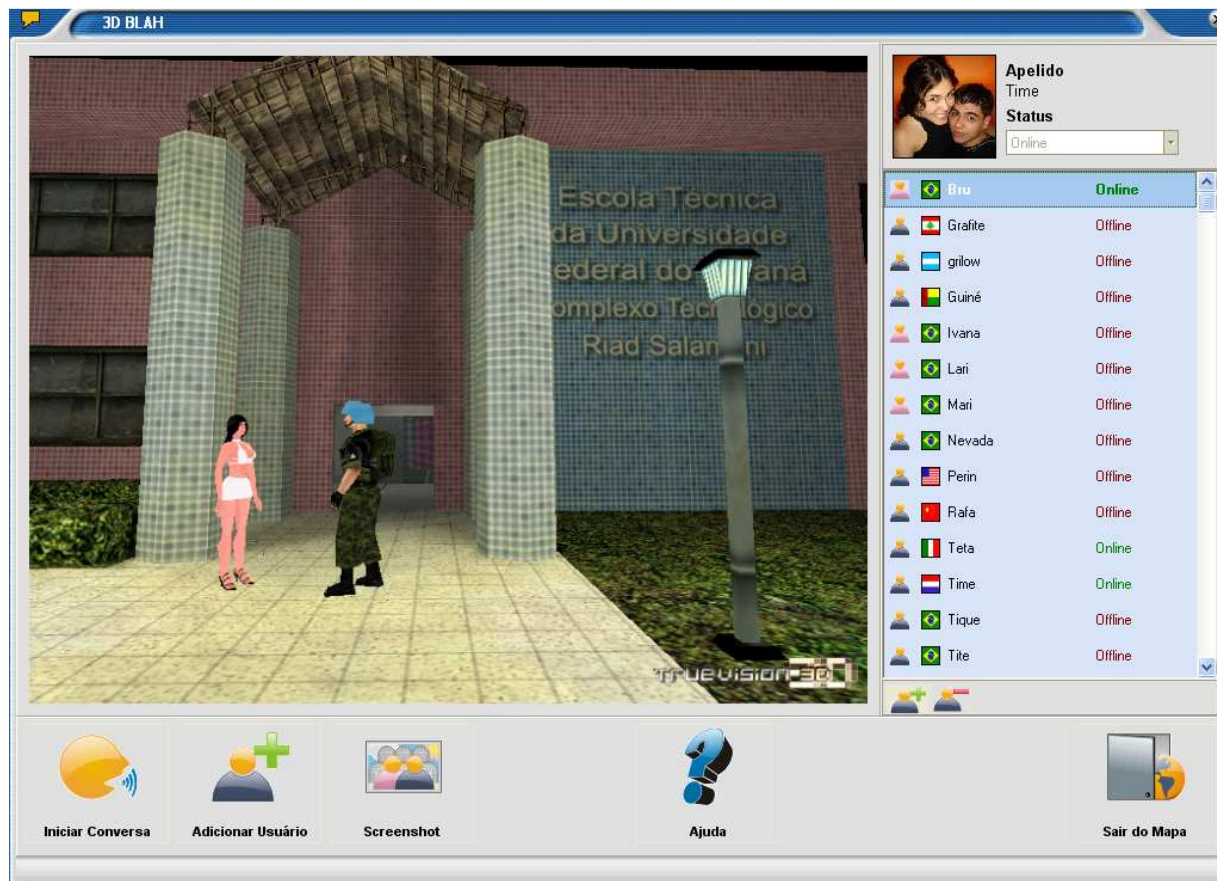
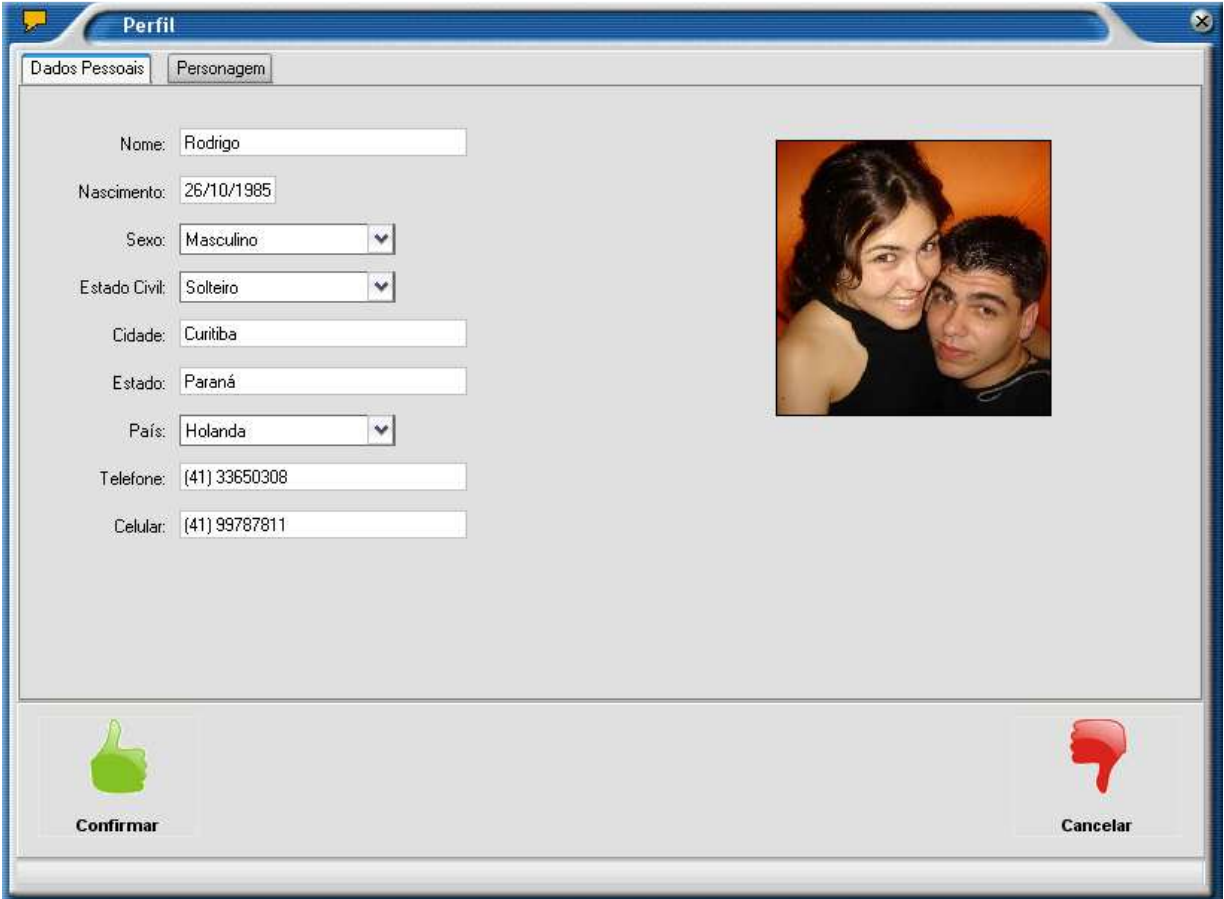


Figura 15 – Tela do mapa do sistema.

Descrição: Esta tela é disposta ao usuário após o mesmo conectar-se a um mapa. Nela estão dispostos o apelido, a lista de contatos ordenada alfabeticamente, o status e a imagem de exibição do usuário, juntamente com os botões Iniciar Conversa (quando o usuário estiver a uma certa distância de um outro modelo, é possível abrir a tela de conversa através deste botão), Adicionar Usuário (quando o usuário estiver a uma certa distância de um outro modelo, é possível adicioná-lo à lista de contatos através deste botão, no caso dele ainda não estar na lista de contatos do usuário), Screenshot (através deste botão o usuário tira um Print Screen apenas da visão que ele tiver no mapa), Sair do Mapa (no qual o usuário sai do mapa no qual esta conectado, retornando para a tela principal), Adicionar Contato (no qual o usuário adiciona um

amigo a sua lista) e Apagar Contato (no qual o usuário exclui o contato selecionado de sua lista de contatos).

7.1.2.4 Tela de Perfil (aba Dados Pessoais)



The screenshot shows a web application window titled "Perfil". It has two tabs: "Dados Pessoais" (selected) and "Personagem". The "Dados Pessoais" tab contains the following form fields:

- Nome: Rodrigo
- Nascimento: 26/10/1985
- Sexo: Masculino (dropdown menu)
- Estado Civil: Solteiro (dropdown menu)
- Cidade: Curitiba
- Estado: Paraná
- País: Holanda (dropdown menu)
- Telefone: (41) 33650308
- Celular: (41) 99787811

To the right of the form is a photograph of a man and a woman. At the bottom of the window, there are two buttons: "Confirmar" (with a green thumbs-up icon) and "Cancelar" (with a red thumbs-down icon).

Figura 16 – Tela do perfil do usuário do sistema.

Descrição: Esta tela é disposta ao usuário após ele clicar no botão “Perfil” na tela principal. Nesta aba é possível a edição dos dados pessoais do usuário, que será visualizado pelos outros usuários. Conforme o usuário altera seu país, a bandeira presente na lista de contatos também é alterada. Após a edição dos dados, basta clicar em “Confirmar” para registrar as alterações.

7.1.2.7 Tela de Perfil (aba Personagem)

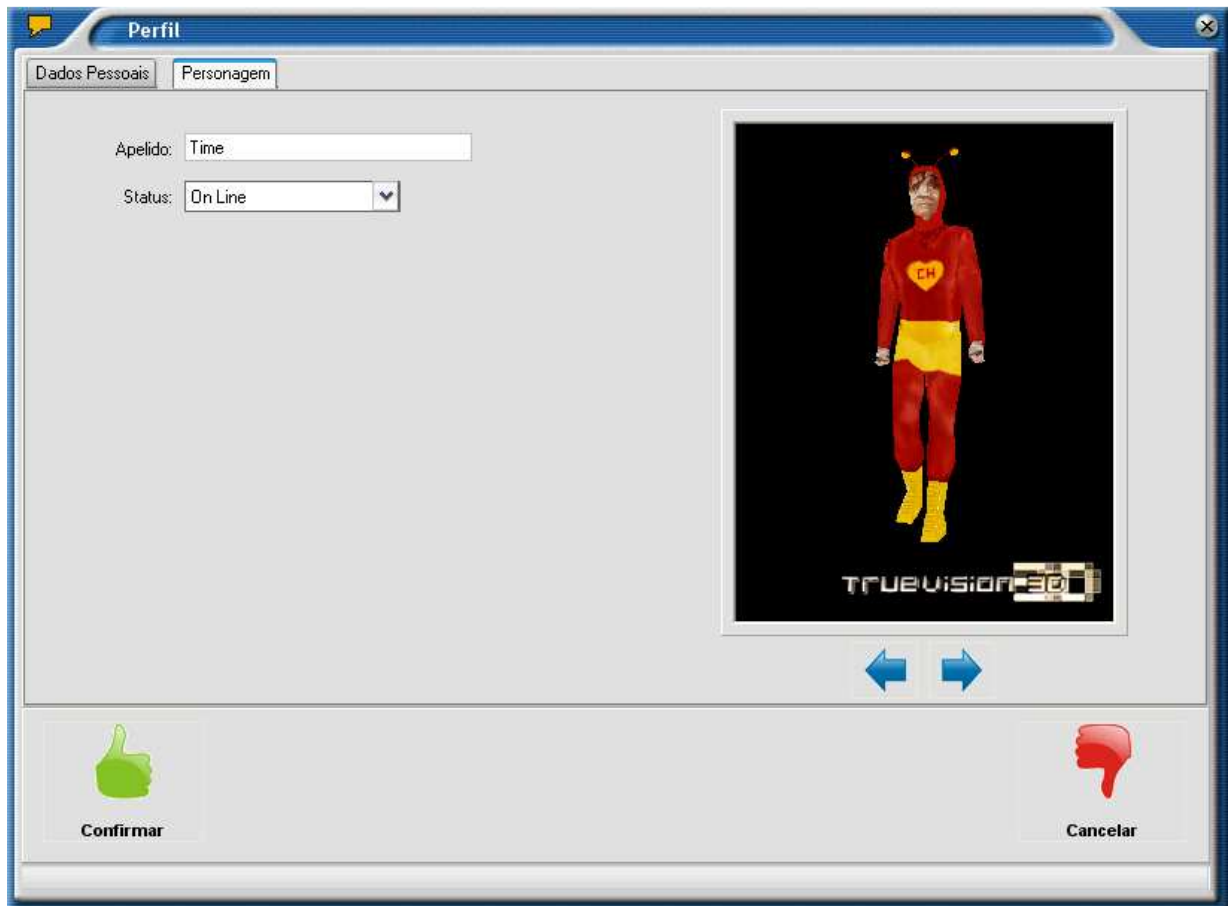


Figura 17 – Tela de escolha do modelo do usuário do sistema.

Descrição: Esta tela é disposta ao usuário após ele clicar no botão “Perfil” na tela principal e selecionar a aba “Personagem” no topo da janela. Nesta aba é possíveis a edição do apelido, status e o avatar usados pelo usuário. Para efetivar as alterações basta clicar em “Confirmar”. É possível a visualização dos modelos disponíveis através das setas de navegação presentes na tela, haverá tantos modelos quanto o usuário tiver adquirido.

7.1.2.6 Tela de Conversa

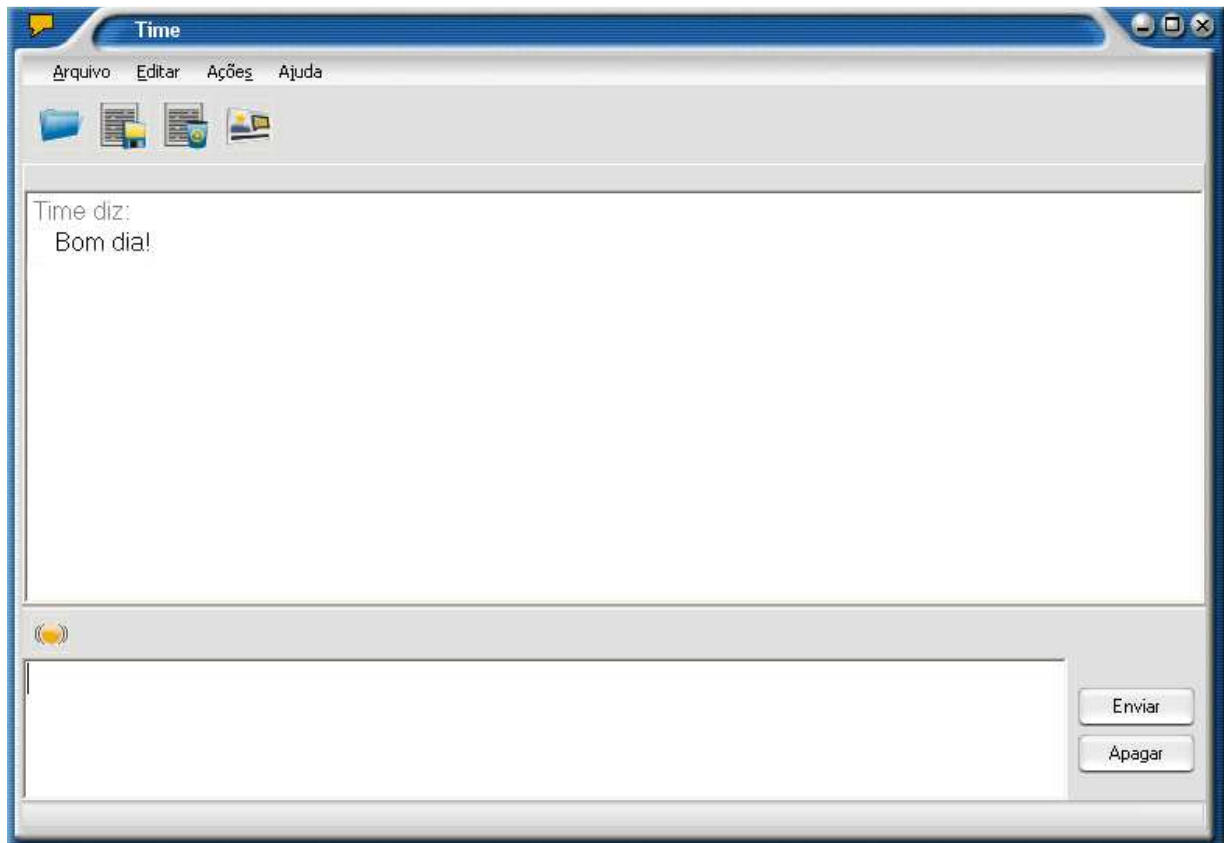


Figura 18 – Tela de conversa entre os usuários.

Descrição: Esta é a tela de conversa, a qual o usuário vai utilizar para comunicar-se com os outros usuários. A tela é dividida basicamente na caixa de conversa, a maior, onde aparecem as mensagens e a caixa de texto, na qual as mensagens são digitadas. Existe ainda, uma barra de ferramentas nela, na qual estão presentes as mesmas ações dos botões dispostos no menu principal, como: “Ver Perfil”, para ver o perfil do usuário; “Limpar Conversa”, para limpar a tela de mensagens (a parte superior em branco); “Salvar Conversa”, para salvar a conversa com seu contato; “Enviar Arquivo”, para enviar algum arquivo ao seu contato; “Nudge”, para chamar a atenção do outro usuário; “Apagar”, para apagar a mensagem digitada; “Enviar”, para enviar a mensagem digitada.

7.1.2.7 Tela de Perfil de outro Usuário



The screenshot shows a window titled "Dados do Contato" with a close button in the top right corner. The window contains a list of contact details for a user named Rodrigo. To the right of the text fields is a small photograph of a man and a woman smiling together.

Nome:	Rodrigo
E-mail:	time
Nascimento:	26/10/1985
Sexo:	Masculino
Estado Civil:	Solteiro
Cidade:	Curitiba
Estado:	Paraná
País:	Holanda
Telefone:	(41) 33650308
Celular:	(41) 99787811

Figura 19 – Tela do perfil do outro usuário.

Descrição: Esta tela é disposta ao usuário quando o usuário através da tela de mensagem clica no ícone de perfil do usuário. Não é possível a alteração desses dados, apenas a visualização.

7.1.2.8 Tela de Configurações (aba Geral)

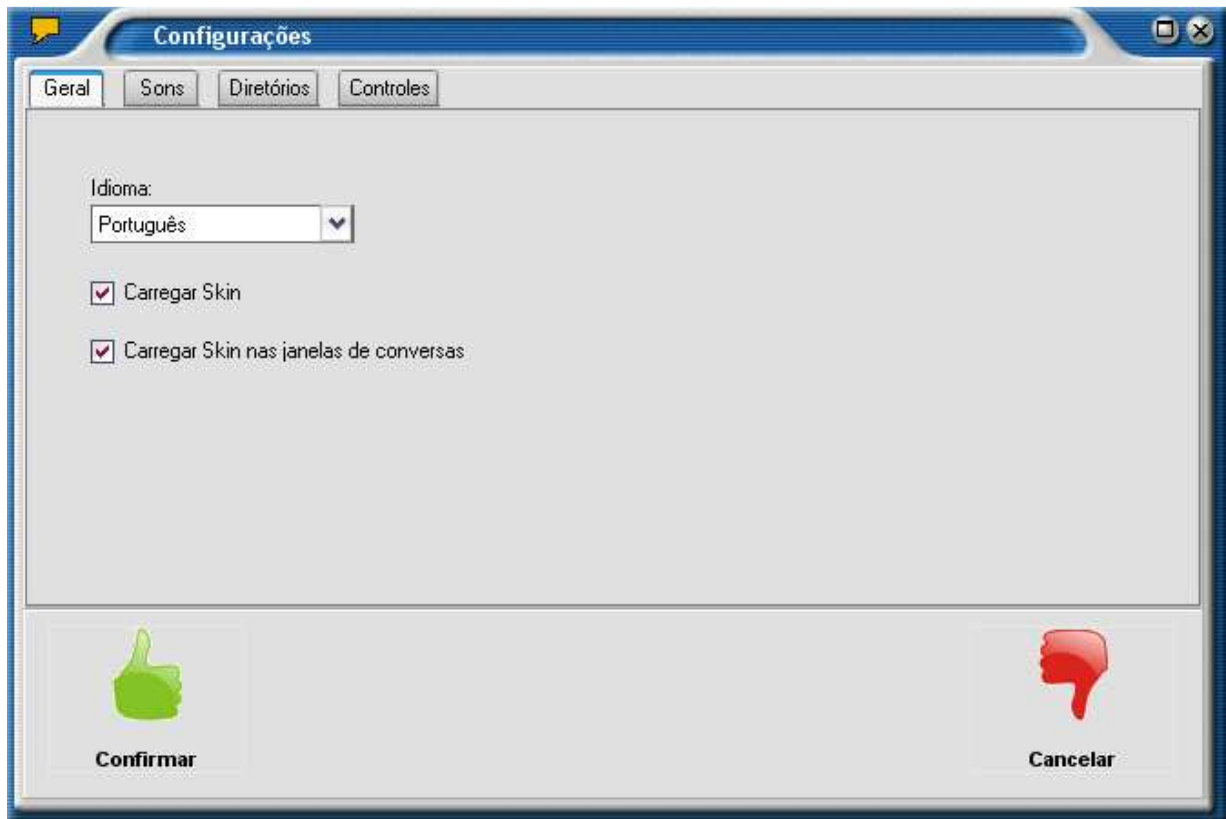


Figura 20 – Tela de configuração do sistema em ambito geral.

Descrição: Para chegar a esta tela o usuário deve clicar em configurações na tela principal, nesta aba é possível a escolha do idioma do aplicativo e se os skins utilizados nas telas serão carregados pelo aplicativo.

7.1.2.9 Tela de Configurações (aba Sons)

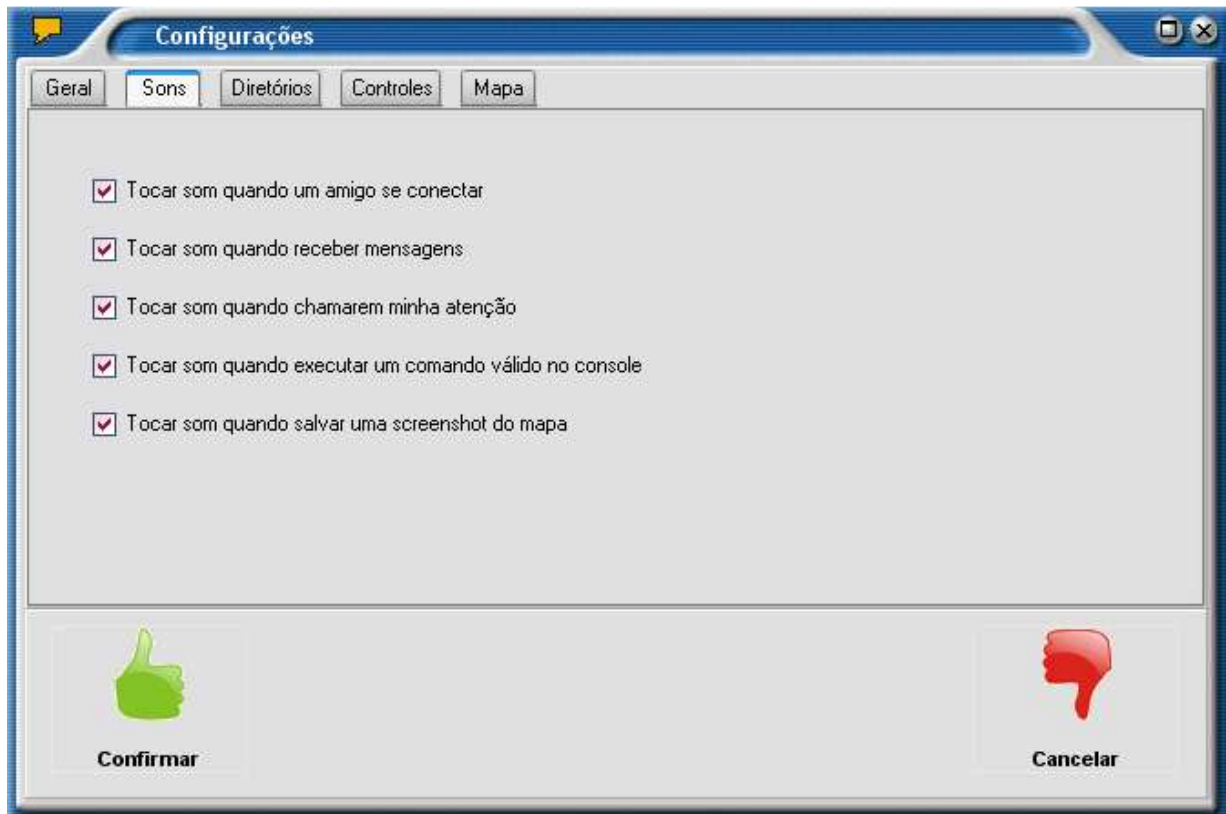


Figura 21 – Tela de configuração dos sons do sistema.

Descrição: Esta tela é referente ao menu de configurações, a aba atual é “Sons”. Nesta parte é possível configurar os sons que você deseja ativar no seu 3DBLAH, como por exemplo, um alerta sonoro quando algum contato seu se conecta no programa.

7.1.2.10 Tela de Configurações (aba Diretórios)

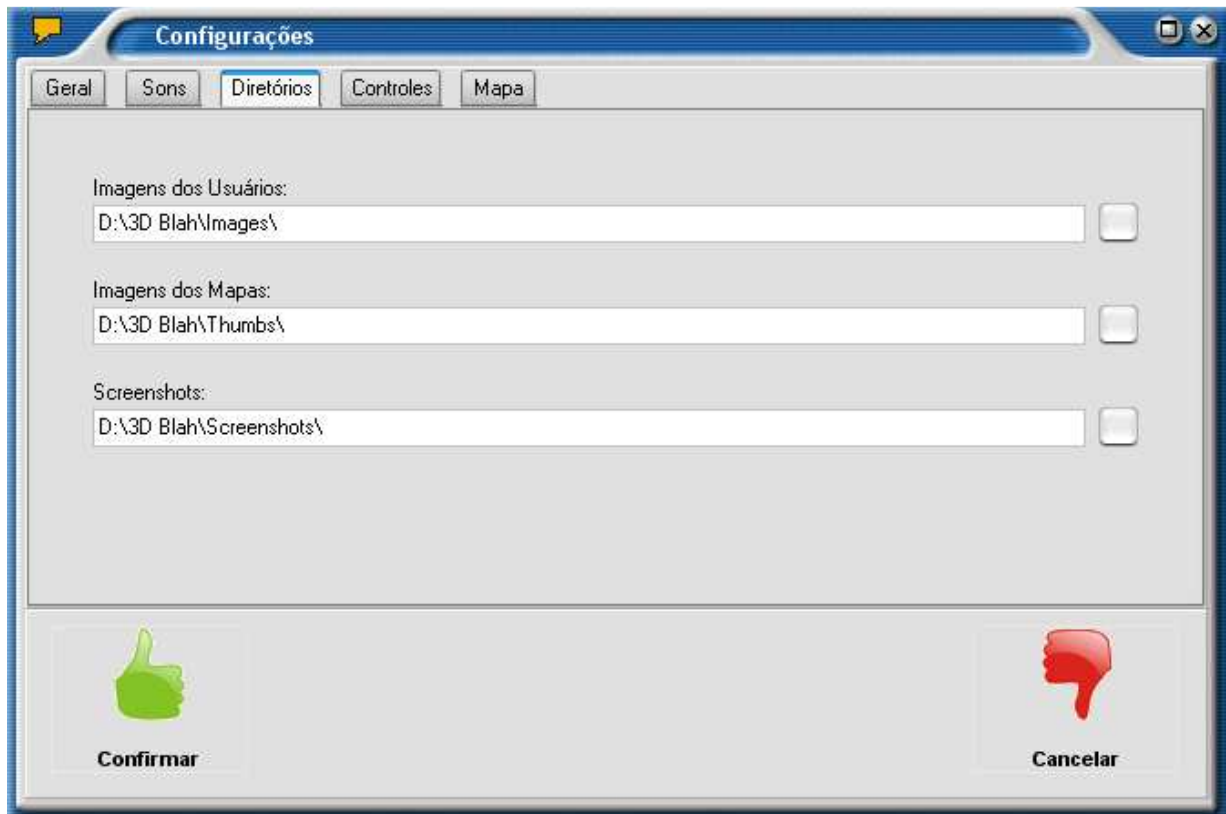


Figura 22 – Tela de configuração de diretórios do sistema.

Descrição: Nesta tela é possível configurar os diretórios nos quais serão salvos como screenshots, imagens dos mapas e imagens dos usuários, podendo ser alterado a qualquer momento, conforme o desejo do usuário.

7.1.2.11 Tela de Configurações (aba Controles)

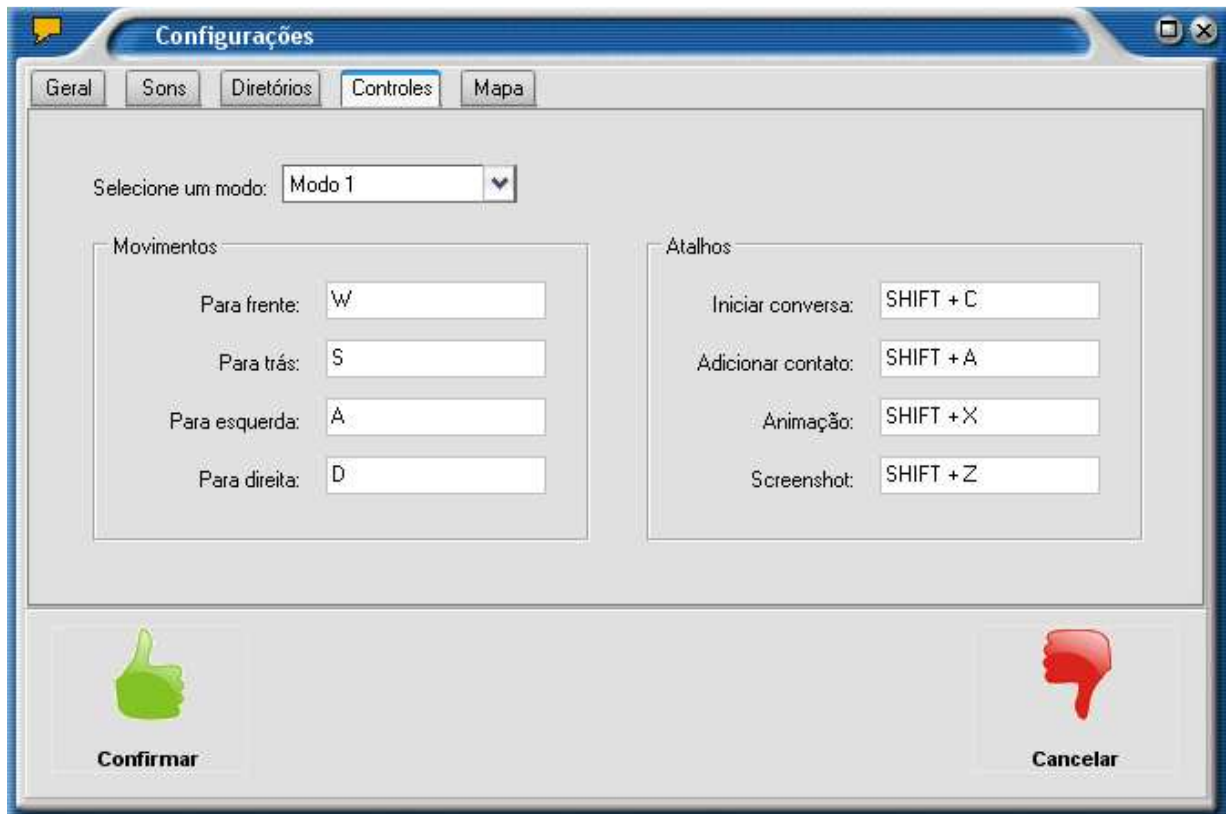


Figura 23 – Tela de configuração dos controles do sistema.

Descrição: Nesta tela é possível configurar os controles que serão usados para movimentar o avatar dentro do cenário 3D. Existem dois modos para escolha de configurações de movimentos.

7.1.2.12 Tela de Configurações (aba Mapas)

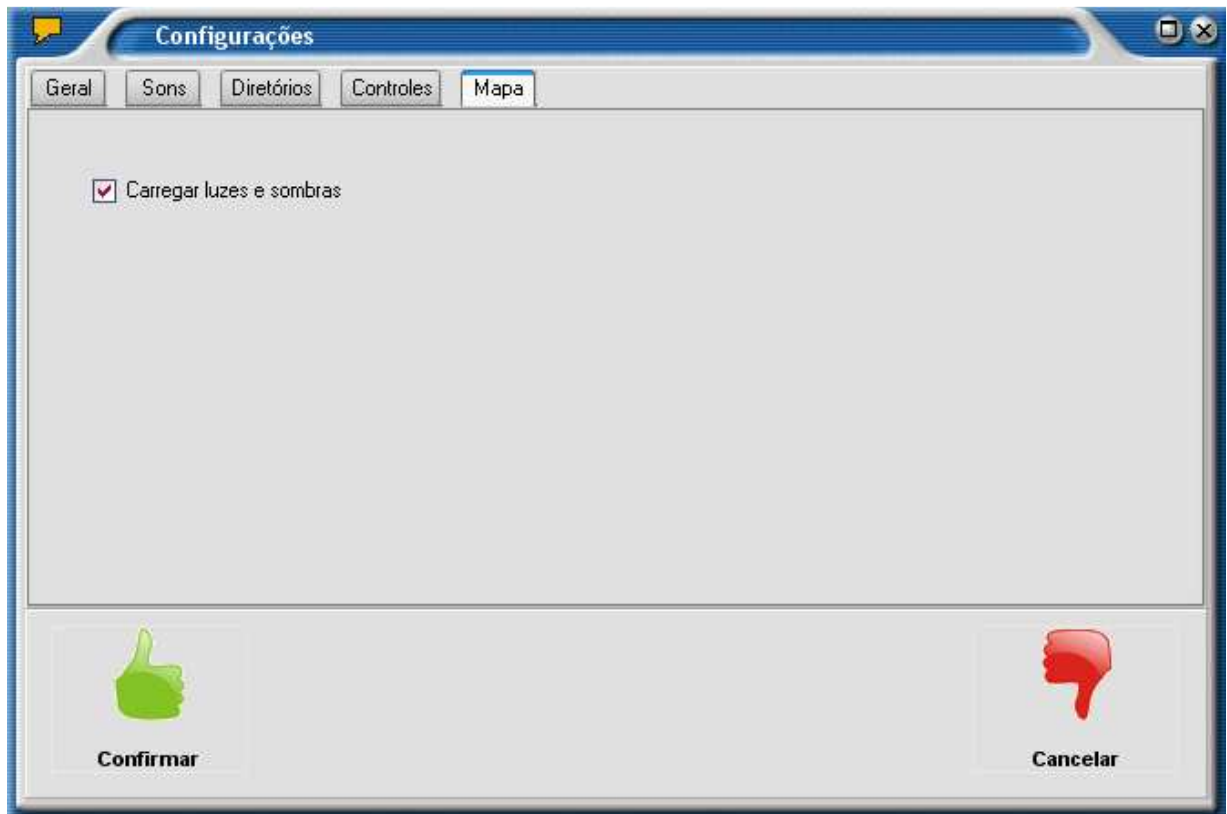


Figura 24 – Tela de configuração dos mapas do sistema.

Descrição: Nesta tela é possível escolher se o usuário deseja os efeitos de luz e sombra dentro do ambiente 3D. Caso opte por não carregar as luzes e sombras, o programa se tornará mais leve, possibilitando um melhor desempenho do programa em máquinas de menor qualidade.

7.1.3 Telas dos Servidores

7.1.3.1 Tela do Servidor de Mapas

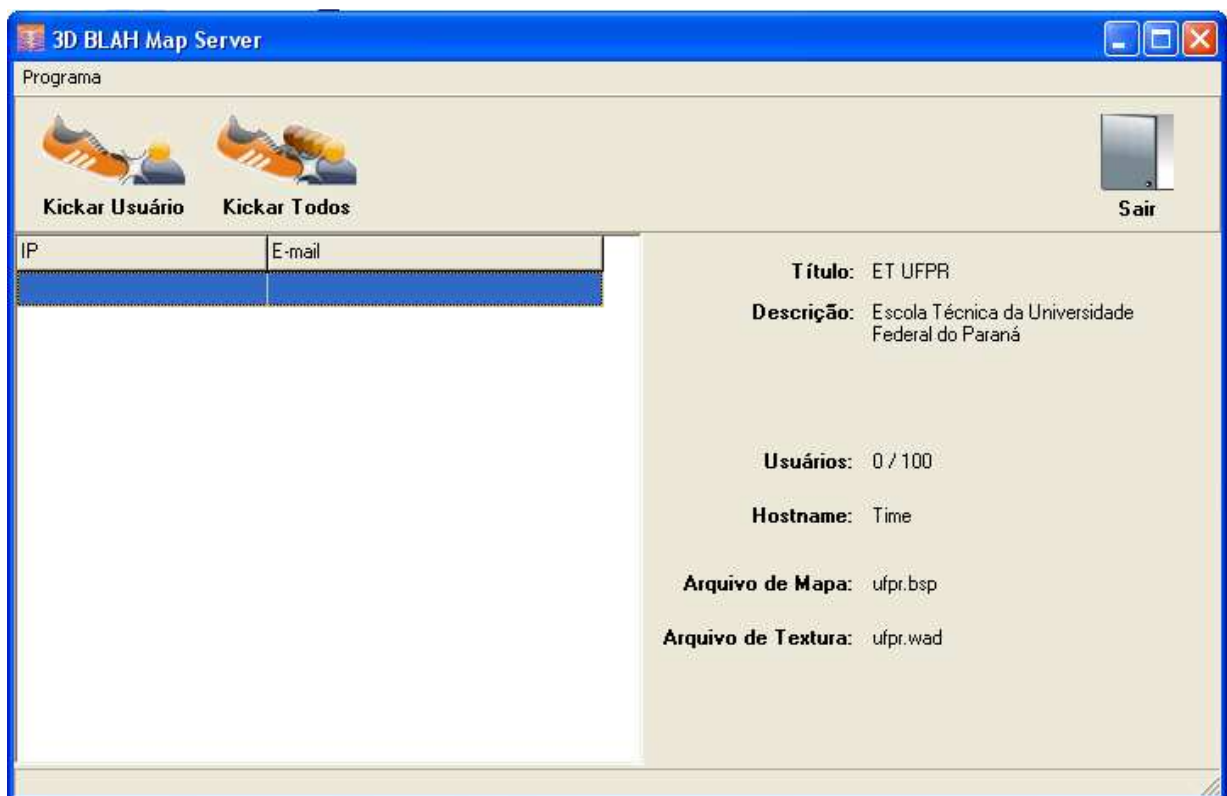


Figura 25 – Tela do servidor de mapa.

Descrição: Apenas o administrador do servidor tem acesso a esta tela. Nela estão dispostos os ip e os e-mails de cada usuário conectado ao mapa que pertence a este servidor. No lado direito estão dados referentes ao mapa, como o nome, capacidade de usuários e a sua descrição. O botão “Kickar Usuário” desconecta o usuário selecionado do mapa, e o botão “Kickar Todos” desconecta todos os usuários presentes no mapa. O botão “Sair” fecha a aplicação, e desconecta todos os usuários conectados ao mapa.

7.1.3.2 Tela Principal do Servidor Geral

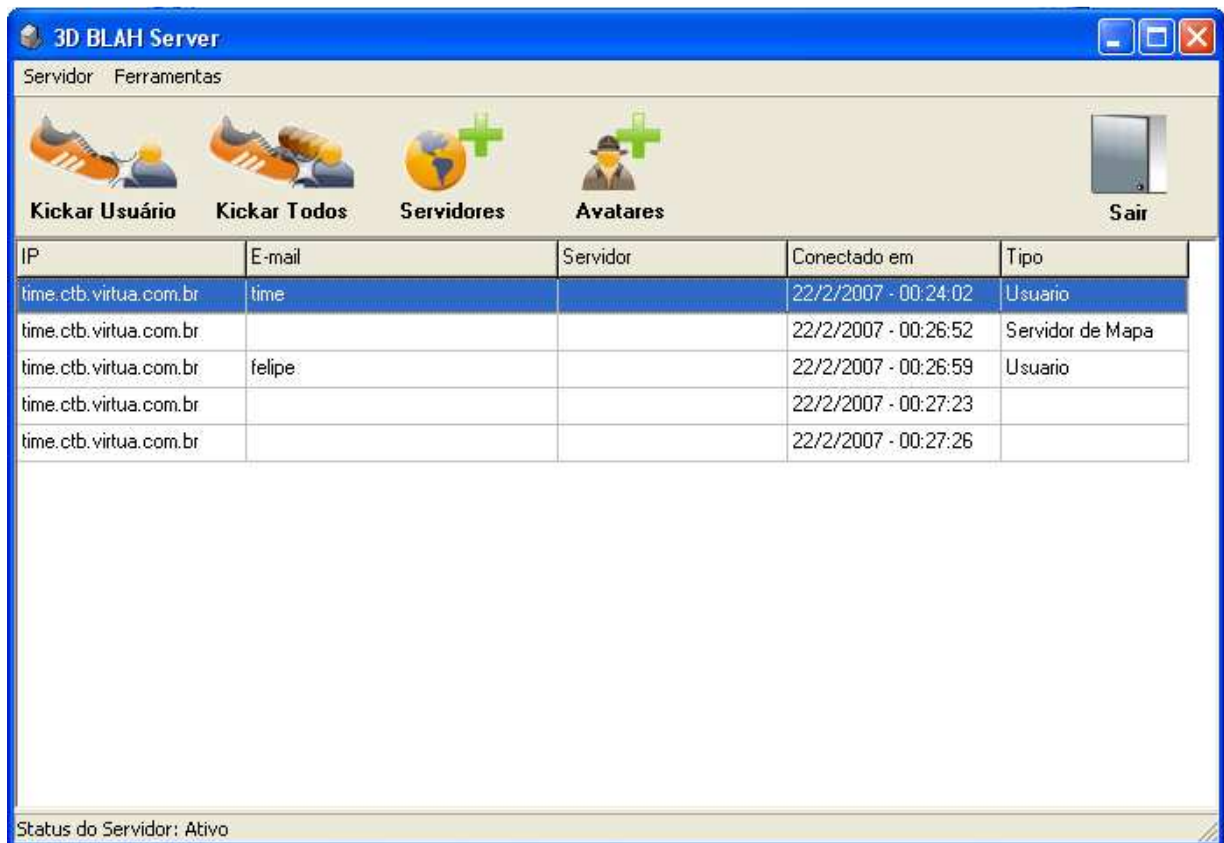


Figura 26 – Tela do servidor principal.

Descrição: Apenas o administrador do servidor tem acesso a esta tela. Nela estão listados todos os usuários conectados no sistema, a data e hora em que se conectou e o tipo de cliente que é, pois o servidor de mapas também é um cliente do servidor principal. O botão “Kickar Usuário” desconecta o usuário selecionado e o botão “Kickar Todos” desconecta todos os usuários. Já os botões “Servidores” e “Avatares”, cadastram novos servidores de mapas e modelos, respectivamente. O botão “Sair” fecha a aplicação, e desconecta todos os usuário conectados.

7.1.3.3 Tela de Cadastro de Servidor de Mapas

Título	Descrição	Hostname
ET UFPR	Escola Técnica da Universidade Federal do Paraná	Time
Sala Fechada	Servidor de testes	Time

Total de Servidores Cadastrados: 2

Figura 27 – Tela de cadastro de mapa no servidor de mapa.

Descrição: Esta tela é disposta ao administrador após ele escolher a opção “Servidores” na tela principal do servidor principal. Nela estão dispostos os botões de navegação na barra de botões, os botões para criação, edição e exclusão de servidores, e botões para salvar o servidor criado ou alterado e sair da tela. Na parte esquerda da tela, estão presentes os campos a serem preenchidos se for criar algum novo servidor, ou então os dados do servidor já existente para alteração, ou ainda os dados para serem apenas visualizados de um servidor existente. Tudo depende da ação do administrador. Do lado direito da tela, existe uma lista com os servidores cadastrados, com alguns dos campos preenchidos no cadastro à disposição.

7.1.3.4 Tela de Cadastro de Modelos

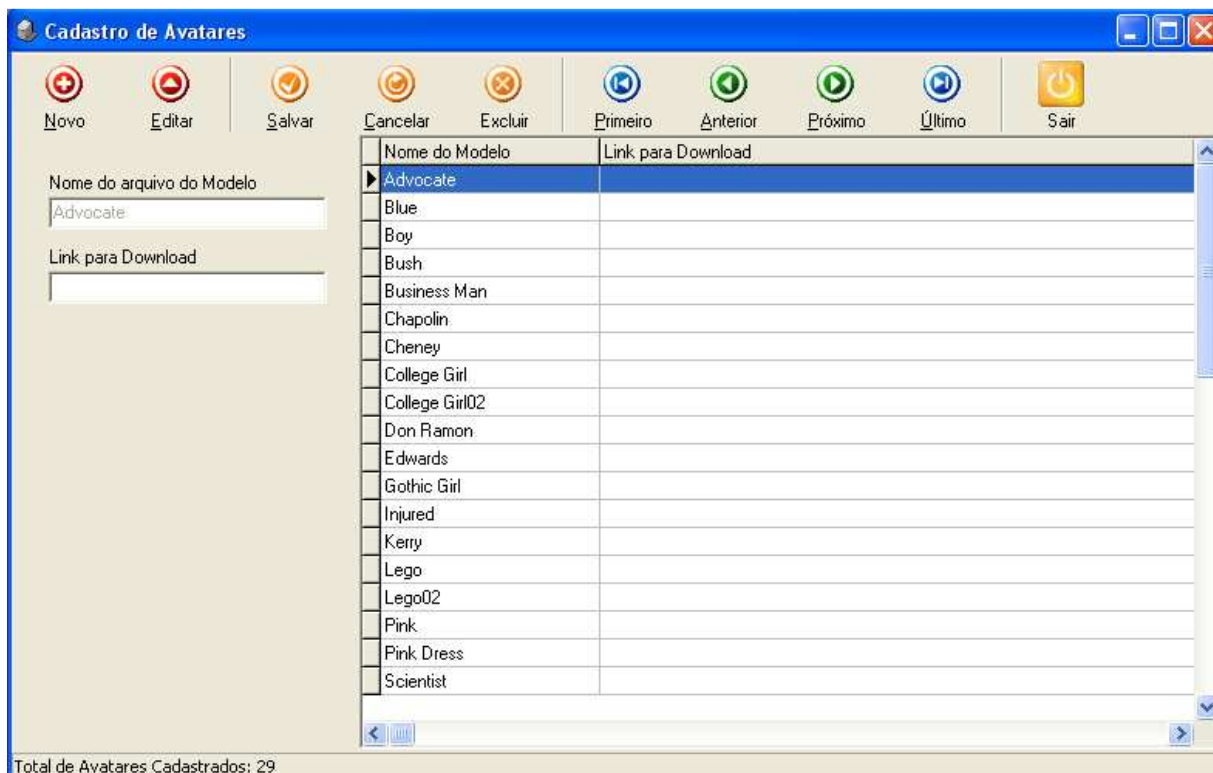


Figura 28 – Tela de cadastro de modelos.

Descrição: Esta tela é disposta ao administrador após ele escolher a opção “Avatares” na tela principal do servidor principal. Nela estão dispostos os botões de navegação, os botões para criação, edição e exclusão de avatares, e botões para salvar o modelo criado ou alterado e um botão para sair da tela. Na parte esquerda existe o campo para o nome do modelo a ser editado/adicionado e o link para o download do mesmo. Do lado direito, estão listados todos os avatares cadastrados no banco de dados pelo administrador.

7.2 Diagrama de Classe

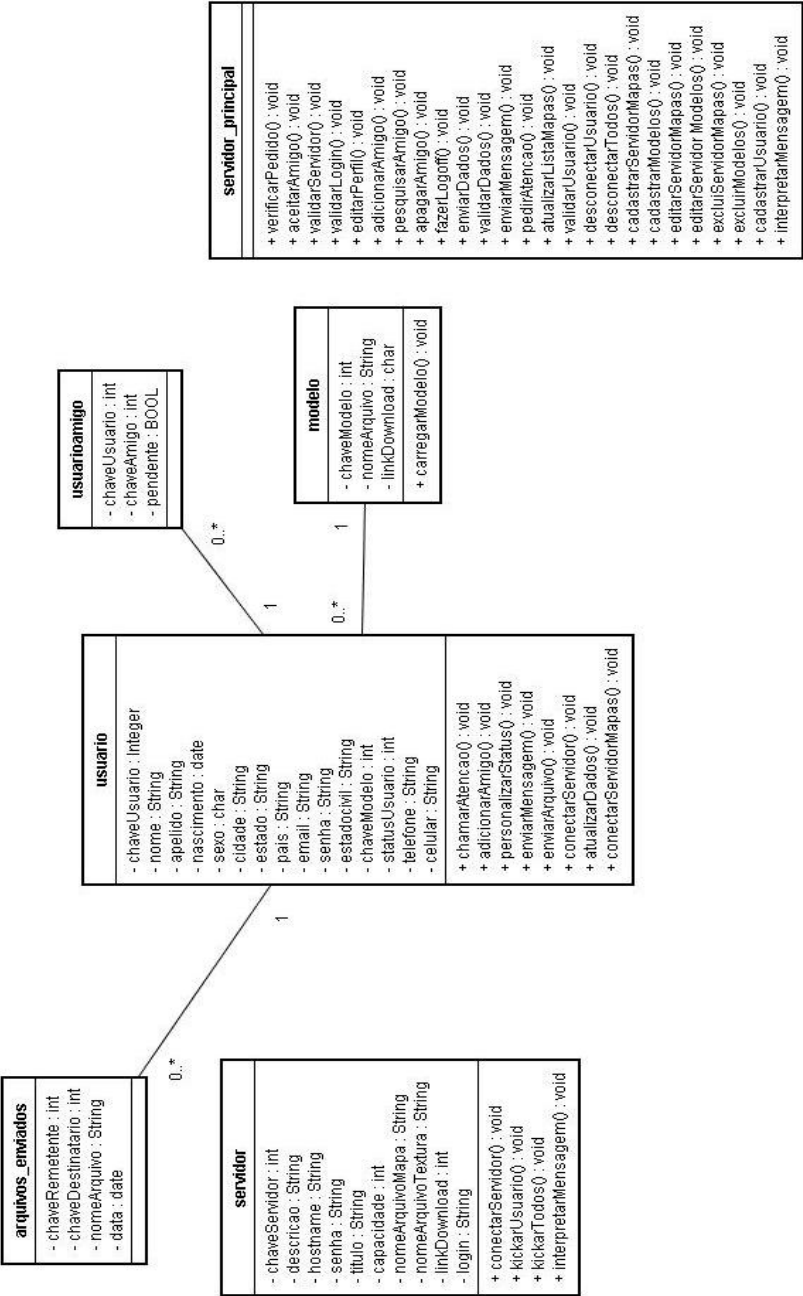


Figura 29 – Diagrama de Classes.

7.3 DER Físico

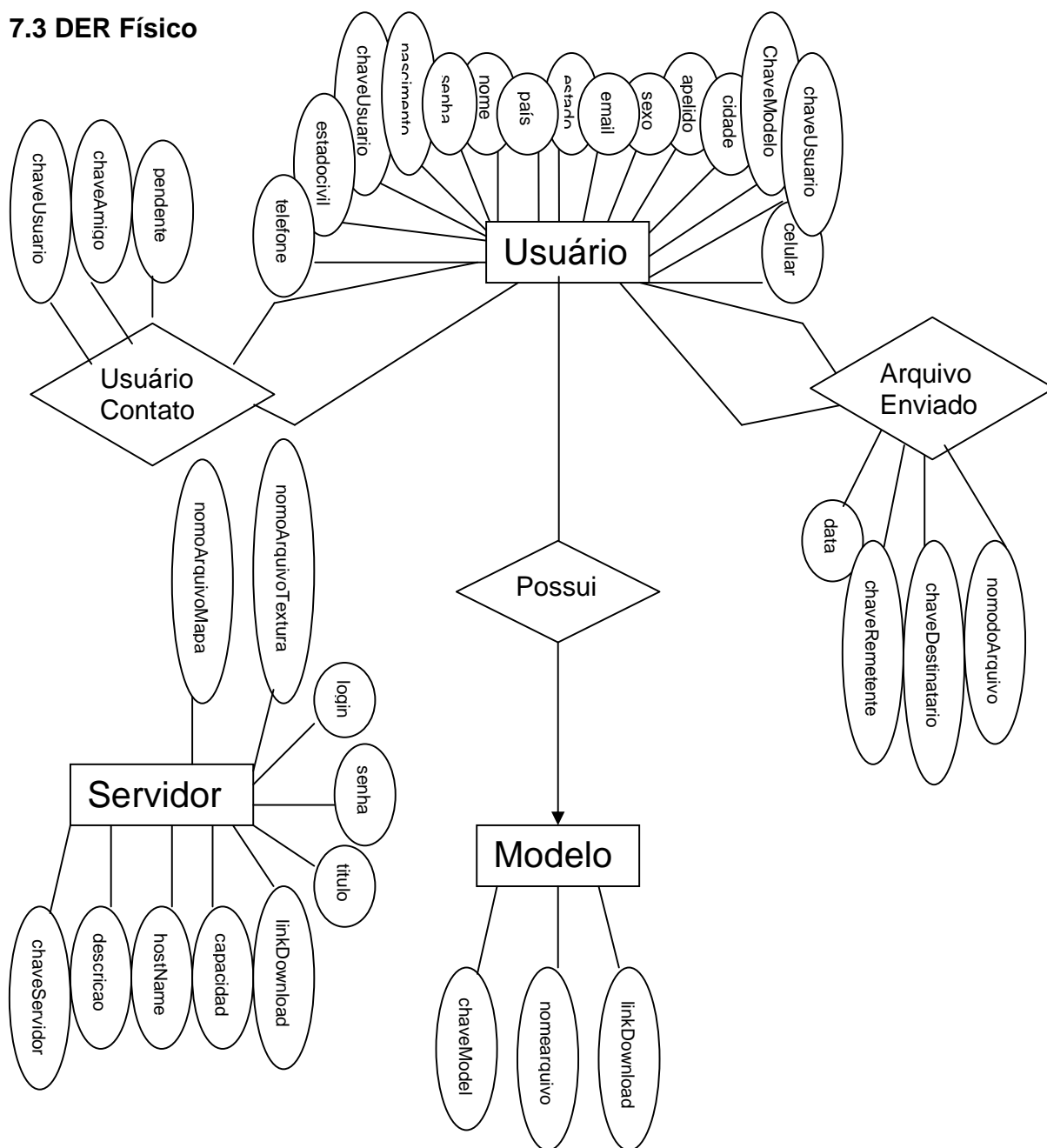


Figura 30 – Diagrama de Entidade Relacionamento.

7.4 Diagramas de Caso de Uso

7.4.1 Do Aplicativo e Servidores

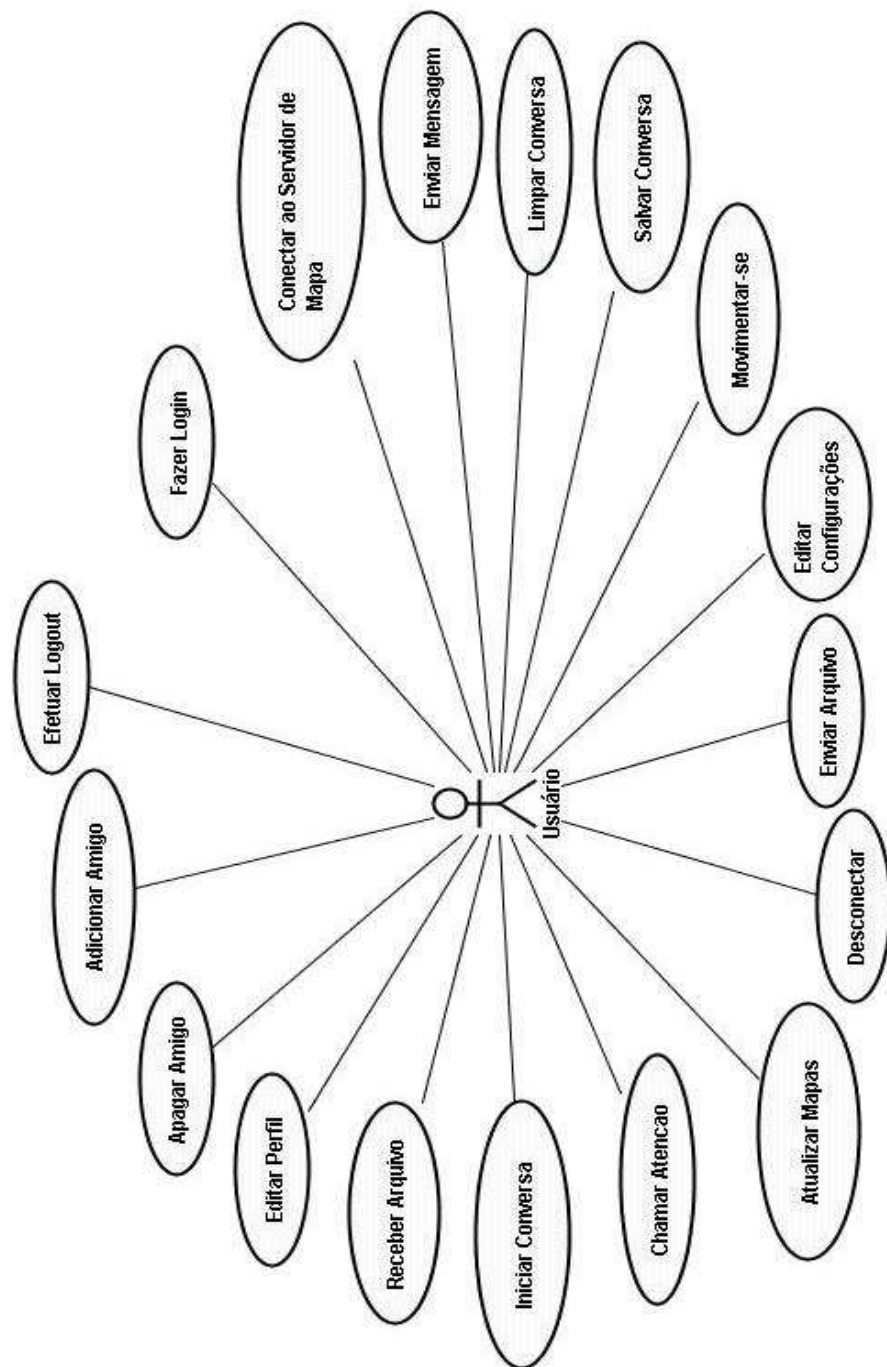


Figura 31 – Diagrama de Caso de Uso do usuário.

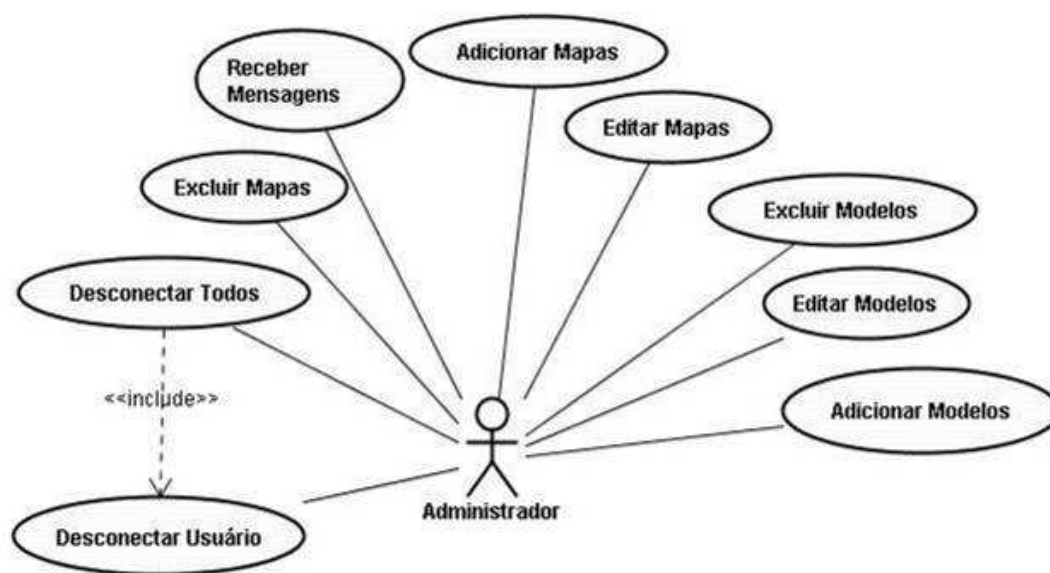


Figura 32 – Diagrama de Caso de Uso do servidor principal.

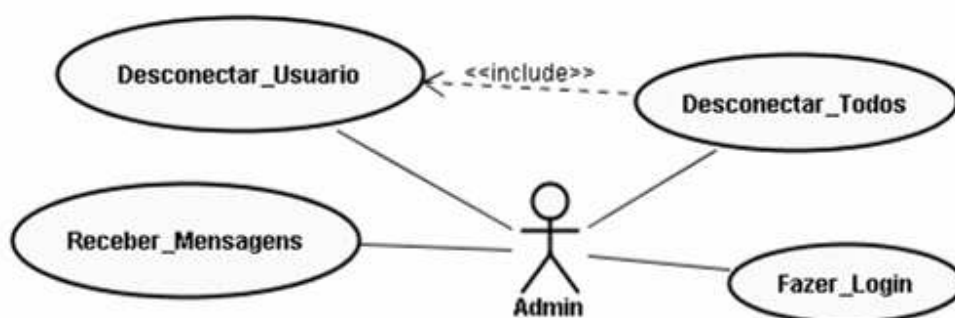


Figura 33 – Diagrama de Caso de Uso do servidor de mapas.

7.4.2 Do Site

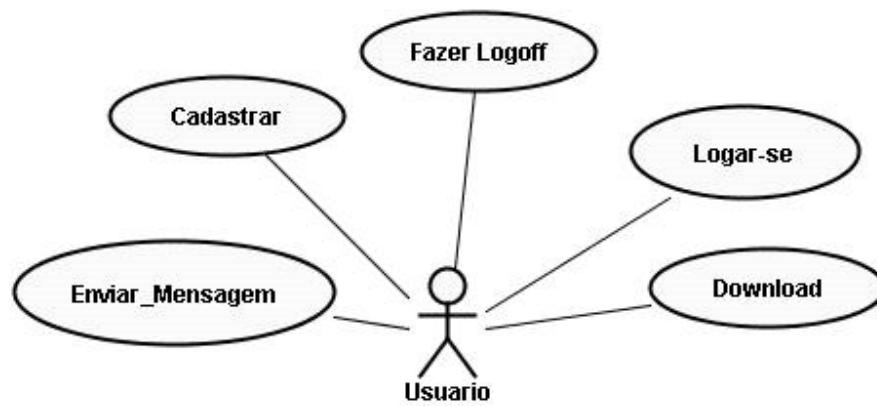


Figura 34 – Diagrama de Caso de Uso usuário em relação ao site.

7.5 Diagrama de Estados do Servidor

Estados do Servidor

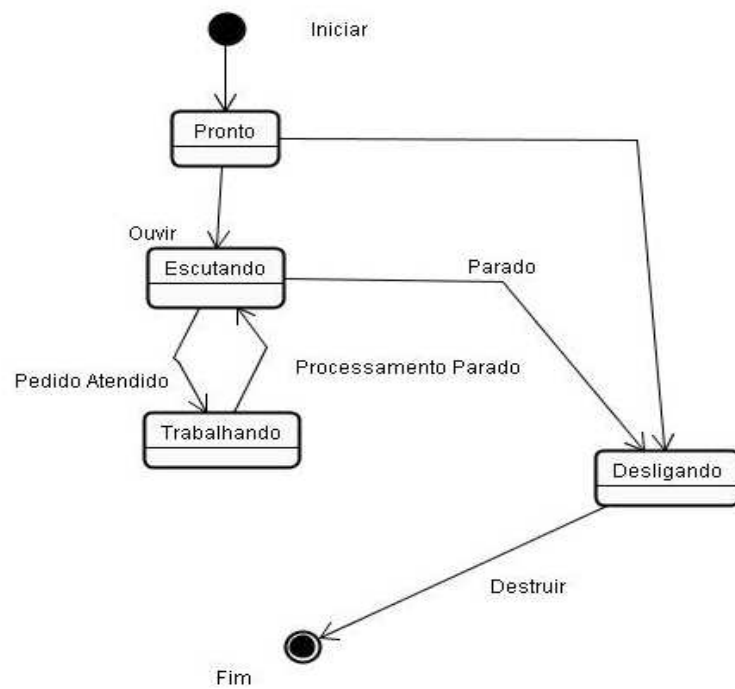


Figura 35 – Diagrama de Estado do Servidor.

Os servidores permanecem ligados escutando os clientes enquanto houver clientes conectados, quando existe algum protocolo ele trabalha, realiza as ações decorrentes de cada protocolo, se ficar parado, o servidor é desligado. Este diagrama representa os dois servidores (Servidor Principal, Servidor de Mapas). O Servidor Principal interpreta os seguintes protocolos: Efetuar Login do Usuário (Cód. 01), Atualizar Dados do Usuário (Cód. 05), Enviar Lista de Avatares (Cód. 07), Enviar Lista de Contatos (Cód. 09), Adicionar Contatos pela Chave (Cód. 11), Adicionar Contatos pelo E-mail (Cód. 12), Remover Contatos do Usuário (Cód. 13), Enviar Dados do Contato (Cód. 15), Enviar Lista de Pedidos de Amigos (Cód. 16), Atualizar Status do Usuário (Cód. 19), Encaminhar Mensagens (Cód. 26), Enviar Páginas de Servidores (Cód. 30), Enviar Dados Gerais (Cód. 35), Efetuar Login do Mapa (Cód. 40), Encaminhar Nudge (Cód. 50), Responder ao Ping (Cód. 53), Registrar Envio de Arquivo (Cód. 56), Enviar Lista de Arquivos para Baixar (Cód. 59), Excluir Registros de um Arquivo Enviado (Cód. 63). Já o Servidor de Mapas interpreta os seguintes protocolos: Informar que Novo Cliente se Conectou (Cód. 32), Enviar Lista de Usuários (Cód. 37), Encaminhar Posição do Usuário (Cód. 45). Conferir a Tabela de Protocolos no Anexo II.

7.6 Diagrama de Atividades

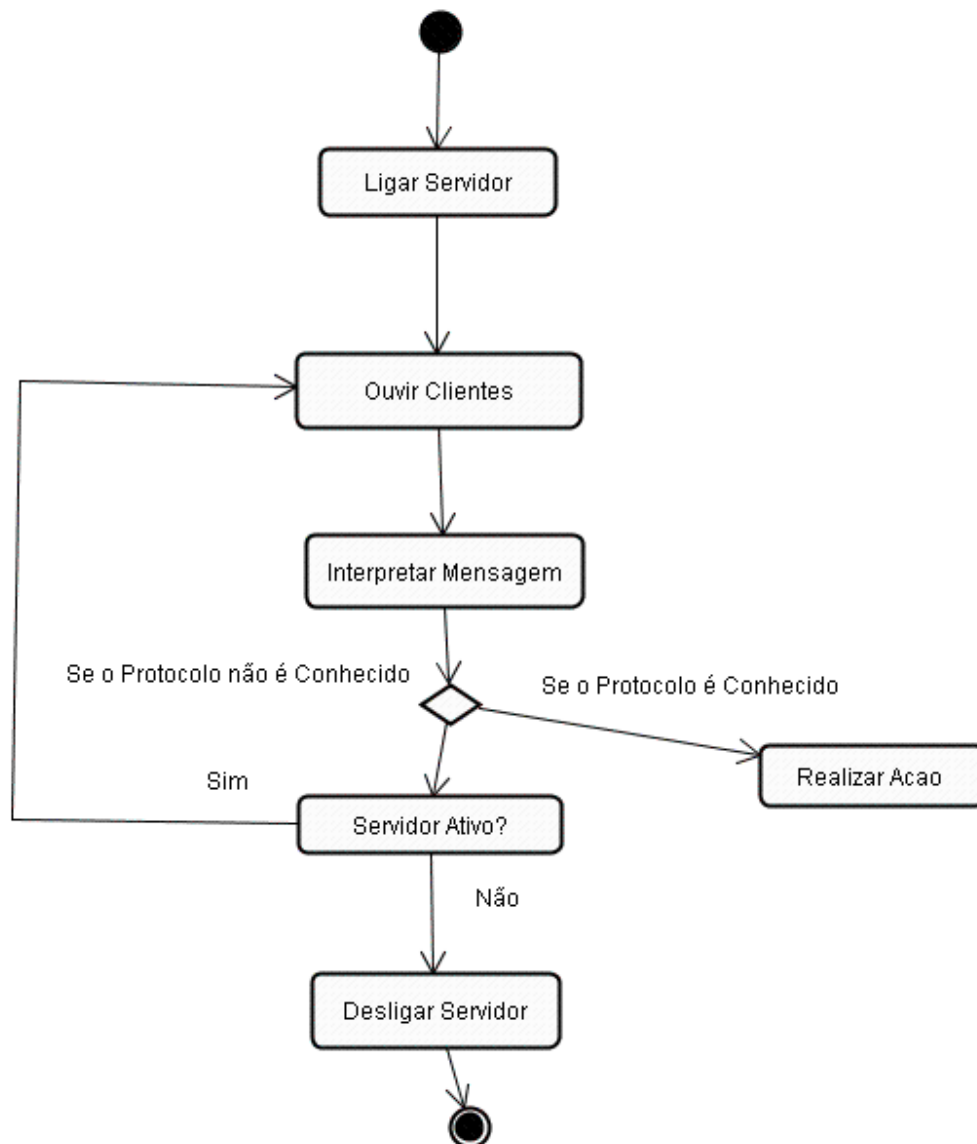


Figura 36 – Diagrama de Atividades.

Neste diagrama estão representadas as atividades do Servidor de Mapas e do Servidor Principal, que efetuam os mesmos processos. Primeiro o servidor é ligado (ou iniciado) e então ele recebe uma mensagem, a interpreta e se a mensagem for conhecida (se o protocolo existir) ele realiza a ação decorrente do protocolo, em caso contrário ele desliga o servidor.

7.7 Descrição de Casos de Usos, Fluxo de Eventos e Diagramas de Seqüência

7.7.1 Casos de Uso do 3DBLAH e dos Servidores

7.7.1.1 C.U.³ Conectar a um Servidor

PRIORIZAÇÃO

Prioridade: Alta

Autor: Rodrigo

Ator: Usuário

Propósito: Conectar a um servidor disponível para utilizar o programa.

Tipo: indispensável

Descrição Resumida

O cliente deve conectar-se ao programa para realizar qualquer ação, inclusive para validar seu login.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso.

Fluxo Principal

- 1 – Cliente manda pedido de conexão com servidor.
- 2 – Servidor é validado.
- 3 – Servidor conecta cliente.
- 4 – Servidor envia resposta ao cliente.

³ C.U. : Sigla para Caso de Uso.

Fluxo Alternativo para o passo 2, caso servidor inválido.

2.1 – Mensagem, “Servidor inválido.”.

2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso servidor indisponível.

2.1 – Mensagem “Servidor Indisponível”.

2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva visita o site 3DBLAH e realiza seu cadastro, após faz o download do programa e o instala. Após ele abre o programa e na interface existem campos para o login e senha, então insere seu login e sua senha e clica em entrar. Enquanto isso o servidor valida a conexão e o login, permitindo ou não a entrada de Silva no sistema.

Diagrama de Sequência

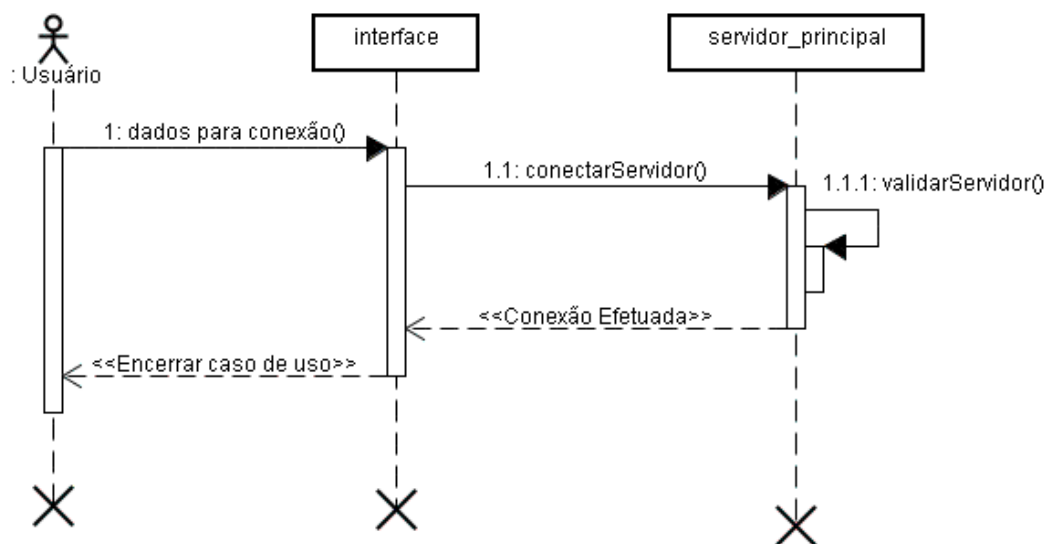


Figura 37 – Diagrama de Seqüência (Conectar ao servidor).

7.7.1.2 C.U. Fazer Login

PRIORIZAÇÃO

Prioridade: Alta

Autor: Gabriel

Ator: Usuário

Propósito: Logar o usuário no sistema.

Tipo: indispensável.

Descrição Resumida

O usuário deve possuir seu cadastro para realizar o login no programa, e após o este realizado é feita a conexão com o servidor, para então usufruir das funcionalidades Presentes no aplicativo.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso.

Fluxo de Eventos

- 1 – Usuário informa dados para logar-se (login e senha).
- 2 – Chama caso de uso CONECTAR A UM SERVIDOR.
- 3 – Servidor valida login.
- 4 – Servidor loga usuário.

Fluxo Alternativo para o passo 3, caso login esteja incorreto.

- 2.1 – Mensagem, “Login Inválido.”.
- 2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 3, caso senha esteja incorreta.

2.1 – Mensagem, “Senha Inválida.”.

2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva visita o site 3DBLAH e realiza seu cadastro, após faz o download do programa e o instala. Após ele abre o programa e na interface existem campos para o login e senha, então ele insere seu login e sua senha e clica em entrar. Enquanto isso o servidor valida a conexão e o login, permitindo ou não a entrada de Silva no sistema.

Diagrama de Seqüência

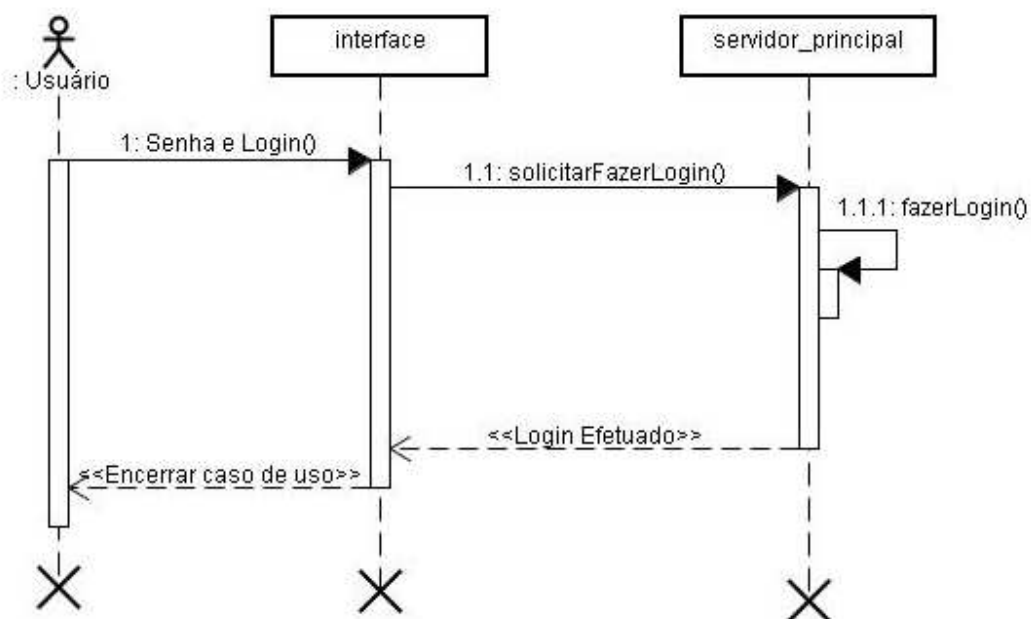


Figura 38 – Diagrama de Seqüência (Fazer Login).

7.7.1.3 C.U. Editar Perfil

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Rafael

Ator: Usuário

Propósito: Editar perfil do usuário.

Tipo: opcional.

Descrição Resumida

O usuário após logado, pode editar seu perfil no programa, editando campos como: apelido, interesses, e-mail, foto, seu avatar e status. Atualizando essas mudanças no Banco de dados simultaneamente.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no sistema.

Fluxo de Eventos

- 1 – Usuário edita perfil.
- 2 – Novo perfil é enviado ao servidor.

Fluxo Alternativo para o passo 1, caso algum campo do perfil esteja incorreto.

- 1.1 – Mensagem, “Não foi possível realizar essa modificação.”.
- 1.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja alterar seu apelido e sua foto de exibição no programa, então clica no ícone “Perfil” na barra de tarefas inferior. Então clica no campo Apelido e realiza a alteração, após clica no botão “Procurar...” para trocar sua imagem de exibição. Uma vez escolhida a imagem, Silva clica no botão “Atualizar”, e estão realizadas as alterações desejadas.

Diagrama de Seqüência

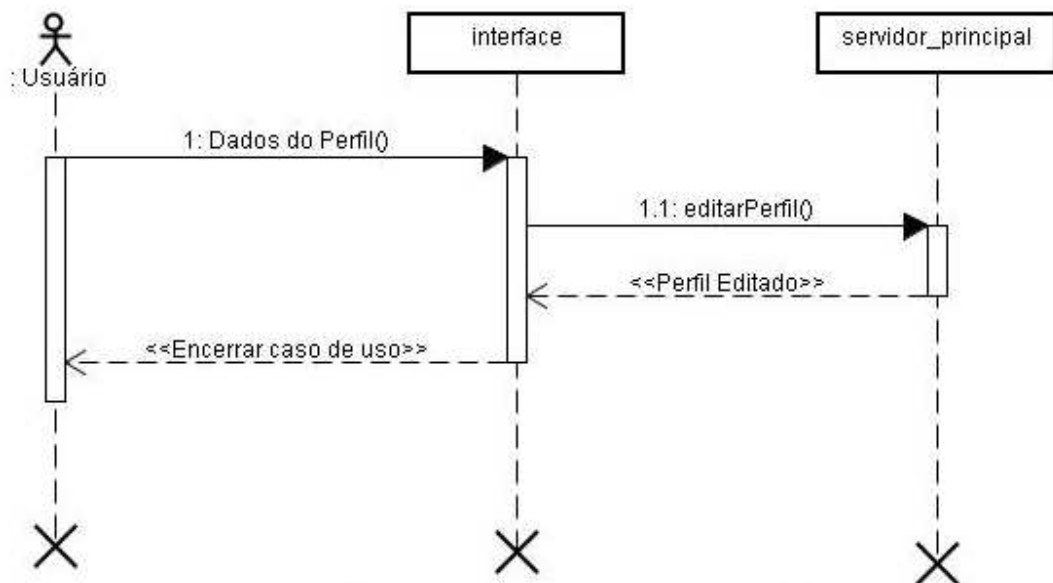


Figura 39 – Diagrama de Seqüência (Editar Perfil).

7.7.1.4 C.U. Editar Configurações

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Rafael

Ator: Usuário

Propósito: Editar configurações do sistema.

Tipo: opcional.

Descrição Resumida

O usuário pode realizar algumas edições, como: carregar *skin* na aplicação, Configurar os controles de movimento, seleccionar os diretórios onde serão salvas as imagens etc.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH, possuir o programa instalado no computador que está tentando o acesso e estar *logado* no programa.

Fluxo de Eventos

1 – Usuário edita configurações.

2 – Sistema realiza edições.

Fluxo Alternativo para o passo 1, caso alguma configuração seja inválida.

1.1 – Mensagem, “Não foi possível realizar essa modificação.”.

1.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja mudar os controles de movimento do personagem, para isso ele clica no ícone “Configurações” no menu inferior, então ele seleciona a aba de configuração de “Controles”. Após ele seleciona o modo 2, na combo de modos de controle. Enfim clica no botão “Confirmar” e realiza suas mudanças.

Diagrama de Seqüência

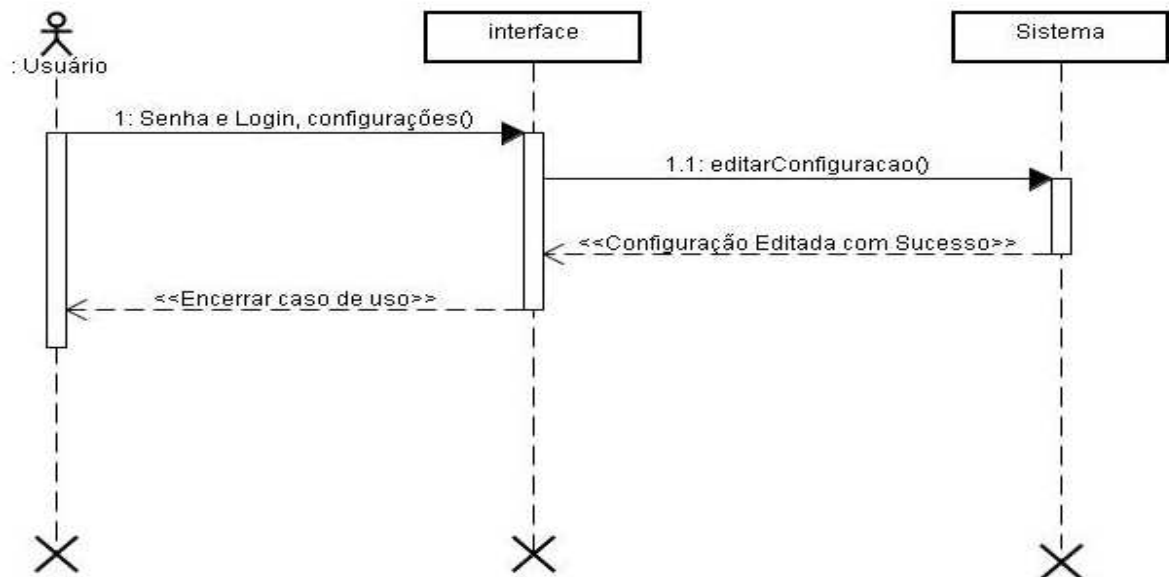


Figura 40 – Diagrama de Seqüência (Editar Configurações).

7.7.1.5 C.U. Adicionar Amigo

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Rafael

Ator: Usuário

Propósito: Adicionar um contato à lista de contatos do usuário.

Tipo: opcional.

Descrição Resumida

O usuário pode adicionar seus amigos e conhecidos à sua lista de contatos, procurando-os ou aceitando pedidos para ser um contato.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no programa e possuir o e-mail do contato a ser adicionado.

Fluxo de Eventos

- 1 – Usuário envia dados do amigo para o servidor.
- 2 – Servidor envia pedido de novo contato ao amigo.

Cenário Principal

O usuário Silva deseja adicionar um amigo através da procura do contato pelo seu e-mail, para isso ele clica em um ícone de um boneco com um "+", após insere o e-mail desejado e clica em pesquisar, após o programa mostra o contato e pergunta se Silva deseja adicionar o amigo, então ele o adiciona e após a aceitação do contato de Silva, eles são inseridos na lista de contatos um do outro.

Diagrama de Seqüência

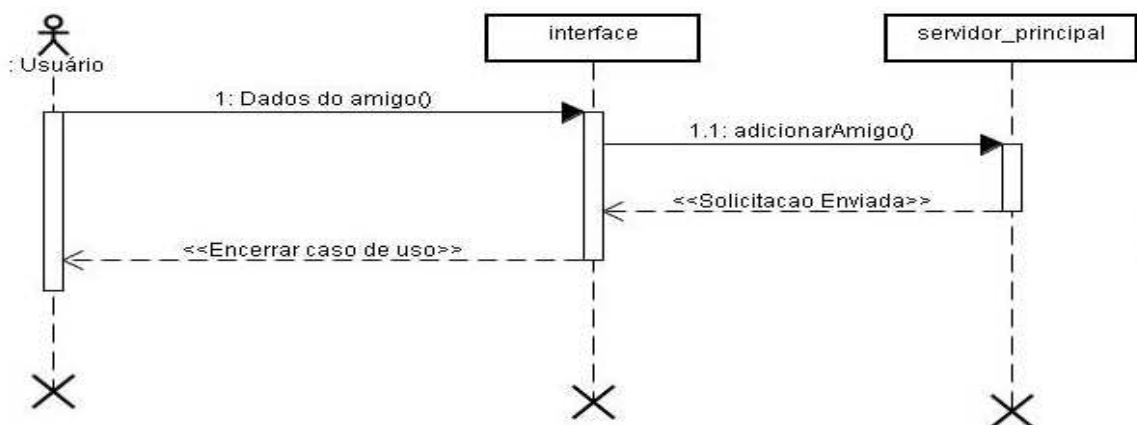


Figura 41 – Diagrama de Seqüência (Adicionar Amigo).

7.7.1.6 C.U. Aceitar Amigo

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Rafael

Ator: Usuário

Propósito: Aceitar e adicionar amigo como contato.

Tipo: opcional.

Descrição Resumida

O usuário pode adicionar seus amigos e conhecidos à sua lista de contatos, procurando-os ou aceitando pedidos para ser um contato.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no programa e existir um pedido de adição de contato para o usuário.

Fluxo de Eventos

- 1 – Servidor envia novo(s) pedido(s) de contato(s) para usuário.
- 2 – Usuário aceita pedido(s).
- 3 – Servidor adiciona contato à lista do usuário.

Fluxo Alternativo para o passo 2, caso usuário recuse o pedido de novo contato.

- 2.1 – Mensagem para o amigo, “Amigo recusou pedido.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja aceitar um pedido de cachorro@teste.com.br, então quando ele loga-se ou se estiver on-line no momento do pedido, o sistema direciona à

ele uma mensagem referente ao pedido de adição de contato podendo aceitar ou não este pedido, no caso de aceito um é adicionado à lista de contatos do outro, caso contrário é enviada uma mensagem à ambos dizendo que o pedido foi negado.

Diagrama de Seqüência

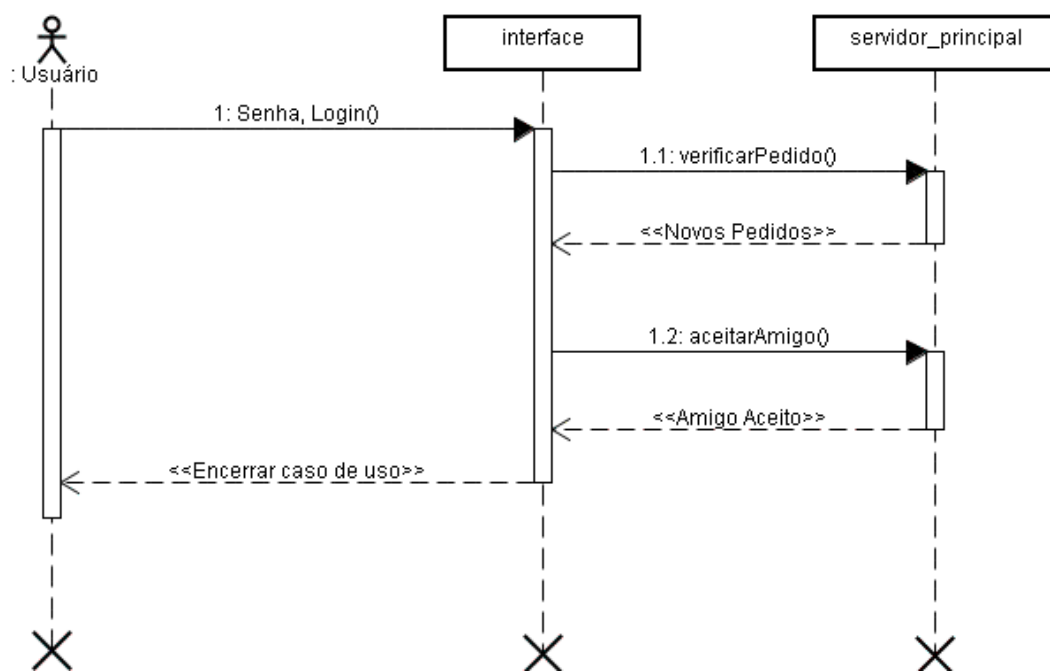


Figura 42 – Diagrama de Seqüência (Verificar Pedido).

7.7.1.7 C.U. Apagar Amigo

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Usuário

Propósito: Apagar amigo da lista de contatos.

Tipo: opcional.

Descrição Resumida

O usuário pode excluir seus amigos e conhecidos de sua lista de contatos, ao excluir um usuário de sua lista de contatos, a mesma é atualizada no banco de dados, sem deletado de lá também.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no programa e possuir um contato para ser excluído.

Fluxo de Eventos

- 1 – Usuário escolhe amigo a ser apagado de sua lista e envia pedido ao servidor.
- 2 – Sistema envia mensagem de confirmação de exclusão ao usuário.
- 3 – Servidor deleta amigo da lista de contatos do usuário.

Fluxo Alternativo para o passo 2, caso usuário não queira deletar amigo.

- 2.1 – Mensagem, “Amigo não excluído.”
- 2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja agora excluir um contato de sua lista, e para isso ele clica no contato uma vez apenas, selecionando-o, após clica no ícone de um boneco com um sinal “-“ no canto direito da tela principal. Então o sistema envia uma mensagem se Silva deseja ou não excluir o contato, em caso positivo, ele é excluído da lista de contatos dele.

Diagrama de Seqüência

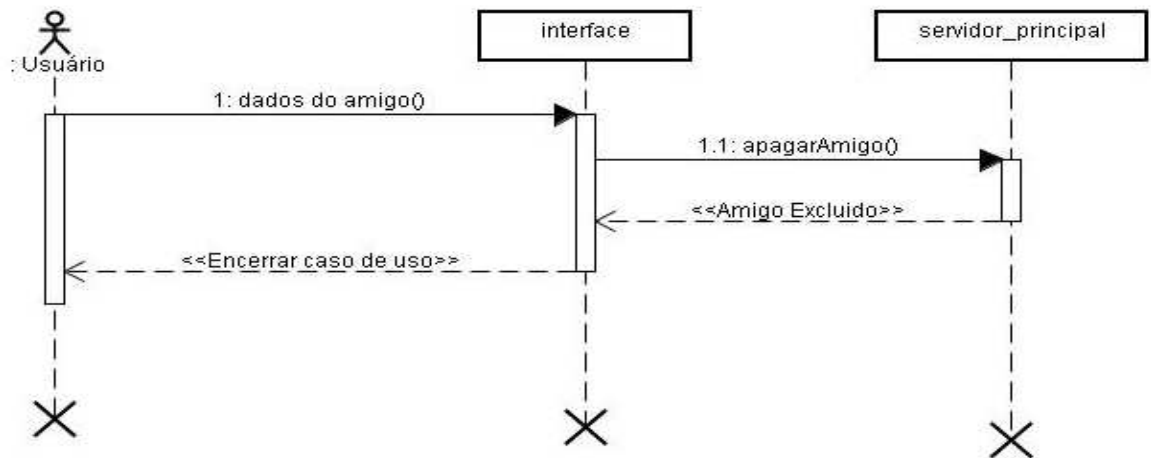


Figura 43 – Diagrama de Seqüência (Apagar Amigo).

7.7.1.8 C.U. Fazer Logoff

PRIORIZAÇÃO

Prioridade: Média

Autor: Felipe

Ator: Usuário

Propósito: Efetuar logout do usuário no servidor.

Tipo: indispensável.

Descrição Resumida

O usuário pode adicionar realizar o Logout, saindo apenas de sua conta sem fechar o programa.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no programa.

Fluxo de Eventos

1 – Usuário efetua logoff.

- 2 – Servidor envia mensagem para confirmação de logoff.
- 3 – Usuário confirma logoff.
- 4 – Servidor desconecta usuário.

Fluxo Alternativo para o passo 3, caso usuário desista de deslogar-se.

3.1 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja agora realizar o logoff de sua conta para entrar com outra conta, então ele clica no botão “Log Off” e após a confirmação na mensagem enviada pelo servidor, ele é desconectado do programa e direcionado a tela de login do programa.

Diagrama de Seqüência

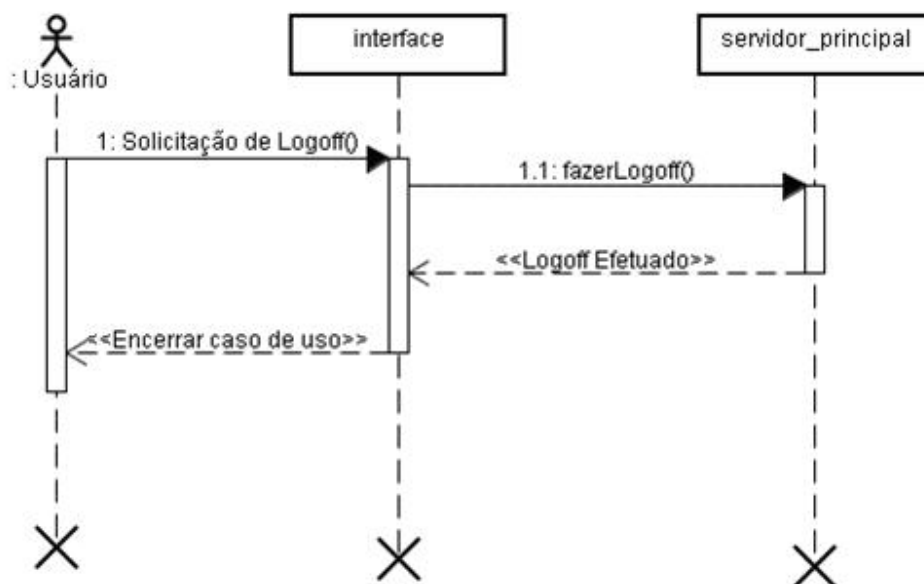


Figura 44 – Diagrama de Seqüência (Fazer Logoff).

7.7.1.9 C.U. Conectar à um Mapa

PRIORIZAÇÃO

Prioridade: Alta

Autor: Felipe

Ator: Usuário

Propósito: Conectar o usuário a um mapa.

Tipo: indispensável

Descrição Resumida

O usuário pode conectar-se em um mapa 3D para exploração ou interação com outros usuários.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está tentando o acesso, estar logado no programa, possuir o mapa em sua máquina e que este mapa esteja rodando no servidor, disponível para conexão.

Fluxo de Eventos

- 1 – Usuário solicita conexão a um mapa.
- 2 – Servidor conecta usuário ao mapa.
- 3 – Servidor envia resposta ao cliente.

Fluxo Alternativo para o passo 2, caso mapa com limite de pessoas excedido.

- 2.1 – Mensagem, “Mapa com limite de Usuários excedido.”.
- 2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso mapa indisponível.

- 2.1 – Mensagem “Mapa Indisponível”.
- 2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja conectar-se a um mapa, para isso ele abre e loga-se no programa, então seleciona um mapa na tela principal e clica em “Conectar-se” ou apenas dá um clique duplo no mapa escolhido, então ele é conectado ao servidor de mapas.

Diagrama de Seqüência

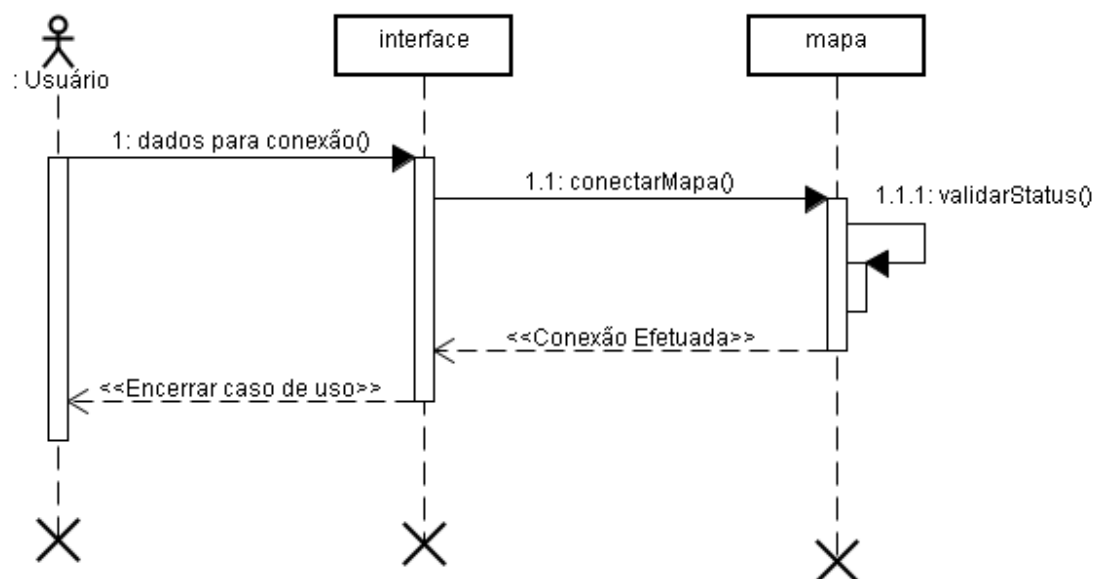


Figura 45 – Diagrama de Seqüência (Conectar ao Mapa).

7.7.1.10 C.U. Estabelecer Conversa

PRIORIZAÇÃO

Prioridade: Alta

Autor: Rodrigo

Ator: Usuário

Propósito: Permitir troca de mensagens entre usuários.

Tipo: indispensável.

Descrição Resumida

O usuário pode iniciar uma conversa com seus contatos ou com outros usuários logados no mesmo mapa que em que o mesmo se encontra.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa, possuir o mapa em sua máquina e que este mapa esteja rodando no servidor, disponível para conexão, e possuir pelo menos um contato na lista de contatos.

Fluxo de Eventos

- 1 – Usuário informa mensagem ao servidor.
- 2 – Servidor envia mensagem ao amigo.

Fluxo Alternativo para o passo 1, caso ocorra algum problema na conexão durante a transferência dos dados.

- 1.1 – Mensagem, “Falha na conexão com o servidor.”.
- 1.2 – Encerra caso de uso.

Cenário Principal

Silva abre o programa, loga-se e percebe que João está *online* em sua lista de contatos, então ele clica duas vezes em cima do nome de João no programa e abre-se a tela de conversa. Após Mariana entra no 3DBLAH, como ela está na lista de Silva, uma janela pop-up aparece para Silva, então este clica nesta janela e inicia uma

conversa com Mariana. Depois de um tempo, Silva conecta-se a um mapa, nele encontra Raimunda que não está em sua lista de contatos, mas deseja conversar com ela, então ele aproxima-se do avatar dela e clica em “Iniciar Conversa”, e a interface de conversa é aberta.

Diagrama de Seqüência

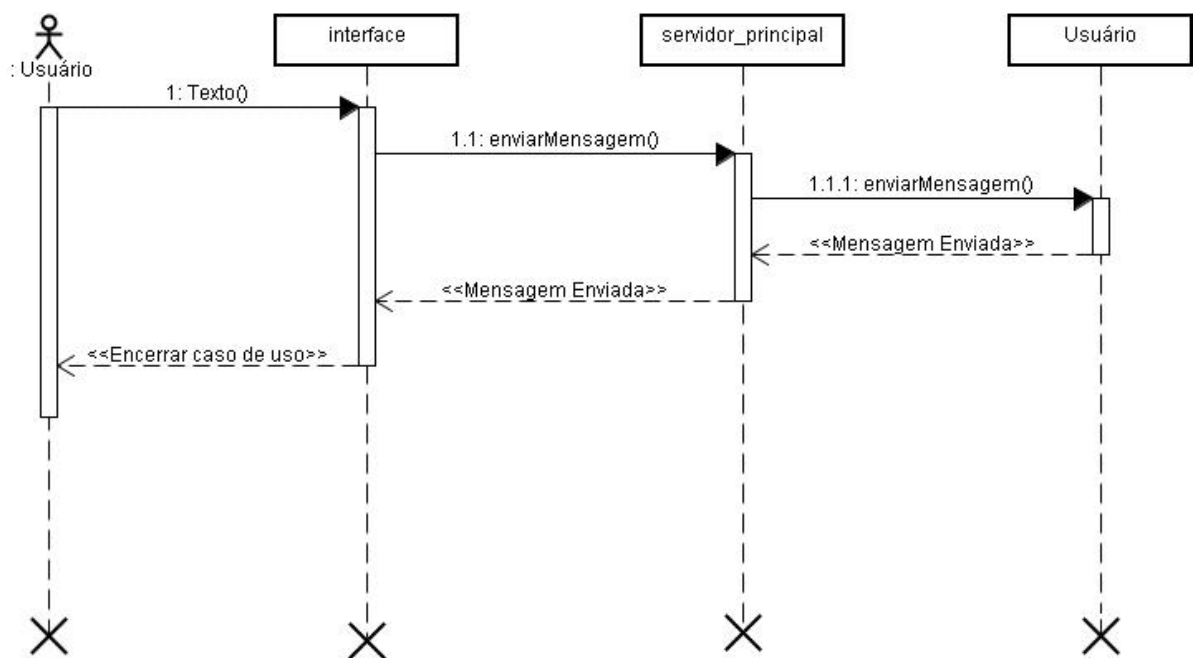


Figura 46 – Diagrama de Seqüência (Enviar Mensagem).

7.7.1.11 C.U. Enviar Arquivo

PRIORIZAÇÃO

Prioridade: Média

Autor: Gabriel

Ator: Usuário

Propósito: Permitir o envio de arquivos entre usuários.

Tipo: opcional.

Descrição Resumida

O usuário pode transferir ao seu contato um arquivo a qualquer momento durante a conversa.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa e possuir pelo menos um contato na lista de contatos com uma conversa estabelecida.

Fluxo de Eventos

- 1 – Usuário informa requisição de envio de arquivo ao servidor.
- 2 – Servidor envia arquivo ao amigo.
- 3 – Servidor recebe confirmação de envio de arquivo.

Fluxo Alternativo para o passo 1, caso ocorra algum problema na conexão durante a transferência dos dados.

- 1.1 – Mensagem, “Falha na conexão com o servidor.”.
- 1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 1, caso o tamanho do arquivo seja maior que o limite permitido.

- 1.1 – Mensagem, “O arquivo desejado é maior que o limite permitido.”.
- 1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso amigo esteja indisponível.

- 2.1 – Mensagem, “Amigo indisponível, tente enviar a mensagem quando ele estiver conectado ao programa.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

Silva deseja transferir um arquivo de texto ao seu contato João, para isso estabelece uma conversa com o mesmo e clica no ícone com uma pasta no canto superior esquerdo da tela de conversa, ou então seleciona a opção Ações na barra de ferramentas, e após a opção “Transferir Arquivo”, então seleciona o arquivo de deseja enviar e clica em “Abrir”.

Diagrama de Seqüência

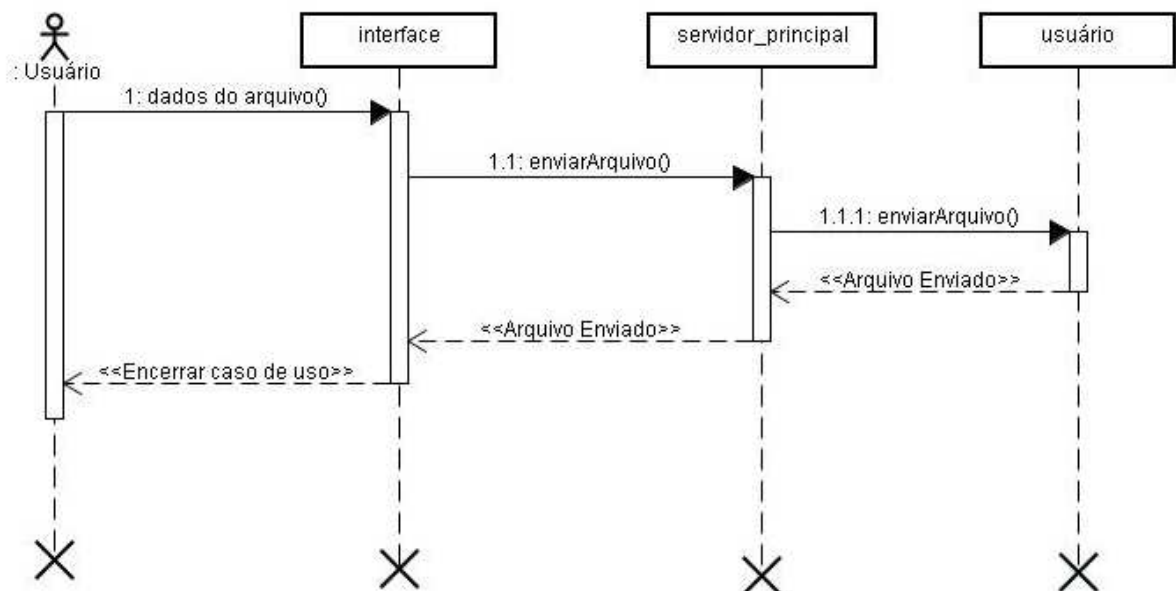


Figura 47 – Diagrama de Seqüência (Enviar Arquivo).

7.7.1.12 C.U. Receber Arquivo

PRIORIZAÇÃO

Prioridade: Média

Autor: Gabriel

Ator: Usuário

Propósito: Permitir o recebimento de arquivos enviados.

Tipo: opcional.

Descrição Resumida

O usuário pode receber de seu contato um arquivo a qualquer momento durante a conversa.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa e possuir pelo menos um contato na lista de contatos com uma conversa estabelecida e um arquivo a ser recebido.

Fluxo de Eventos

- 1 – Servidor informa que há um arquivo a receber.
- 2 – Usuário recebe arquivo.
- 3 – Servidor apaga requisição de arquivo.
- 4 - Servidor apaga arquivo.

Cenário Principal

Silva conecta-se ao programa, e um amigo seu o envia um arquivo, então Silva clica em “Receber”, e o arquivo é recebido pelo usuário após a mensagem de confirmação.

Diagrama de Seqüência

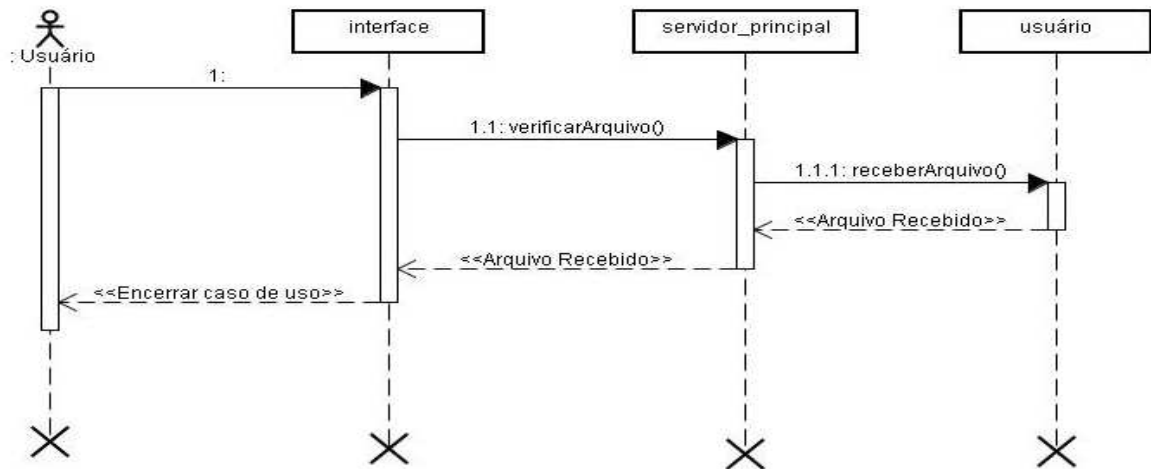


Figura 48 – Diagrama de Seqüência (Receber Arquivo).

7.7.1.13 C.U. Movimentar Modelo

PRIORIZAÇÃO

Prioridade: Alta

Autor: Rodrigo

Ator: Usuário

Propósito: Permitir o movimento do modelo e a exploração do mapa.

Tipo: indispensável.

Descrição Resumida

O usuário pode movimentar seu avatar no ambiente 3D enquanto estiver conectado a um mapa.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa, possuir o mapa em sua máquina e que este mapa esteja rodando no servidor, disponível para

conexão, possuir pelo menos um contato na lista de contatos e possuir um modelo vinculado ao seu perfil.

Fluxo de Eventos

- 1 – Usuário envia os dados do movimento ao mapa.
- 2 – O mapa encaminha posicao do modelo aos outros usuarios.

Fluxo Alternativo para o passo 1, caso ocorra algum problema na conexão durante a transferência dos dados.

- 1.1 – Mensagem, “Falha na conexão com o servidor.”.
- 1.2 – Encerra caso de uso.

Cenário Principal

Silva deseja movimentar seu personagem até um banco localizado no mapa em que está conectado, para isso ele utiliza as teclas de comandos de movimentos e dirige-se ao local escolhido.

Diagrama de Seqüência

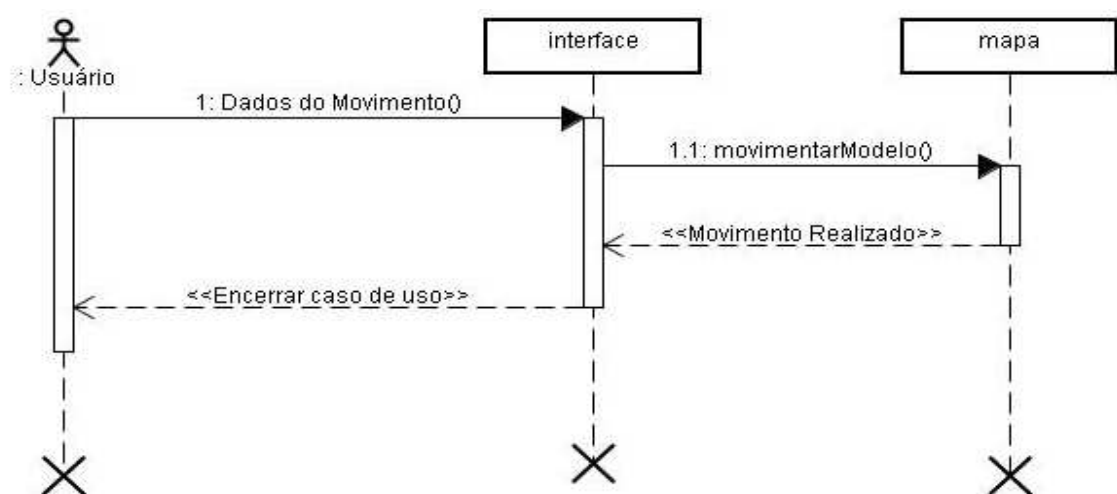


Figura 49 – Diagrama de Seqüência (Movimentar Modelo).

7.7.1.14 C.U. Chamar Atenção

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Usuário

Propósito: Chamar a Atenção dos usuários.

Tipo: opcional.

Descrição Resumida

O usuário pode chamar a atenção de um contato a qualquer momento através do botão “Nudge”.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa e possuir pelo menos um contato na lista de contatos com uma conversa estabelecida.

Fluxo de Eventos

- 1 – Usuário informa requisição de chamado de atenção ao servidor.
- 2 – Servidor chama atenção ao amigo.
- 3 – Cliente recebe confirmação do servidor através de som e deslocamento da tela de conversa.

Fluxo Alternativo para o passo 1, caso ocorra algum problema na conexão durante o pedido de chamado de atenção.

- 1.1 – Mensagem, “Falha na conexão com o servidor.”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso amigo esteja indisponível.

2.1 – Mensagem, “Amigo indisponível, tente enviar a mensagem quando ele estiver conectado ao programa.”.

2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 3, caso cliente não recebe confirmação do servidor.

3.1 – Mensagem, “Verifique a sua conexão com o servidor.”.

3.2 – Encerra caso de uso.

Cenário Principal

Silva deseja chamar a atenção de seu contato João, para isso estabelece uma conversa com o mesmo e clica no ícone “Nudge” com um círculo e ondas em torno dele, localizado na parte meridional da tela.

Diagrama de Seqüência

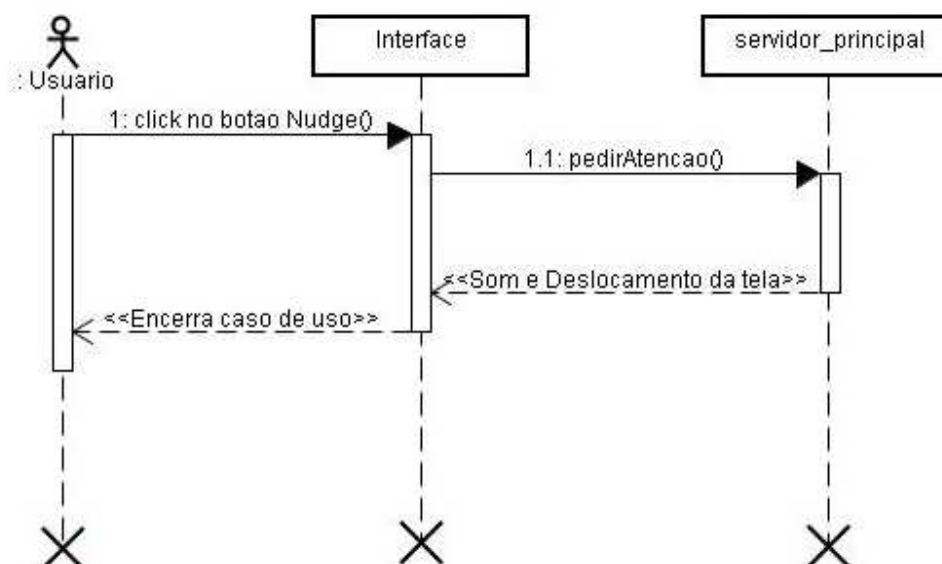


Figura 50 – Diagrama de Seqüência (Pedir Atenção).

7.7.1.15 C.U. Atualizar Tela

PRIORIZAÇÃO

Prioridade: Média

Autor: Gabriel

Ator: Usuário

Propósito: Usuário pode atualizar a tela principal e verificar os mapas disponíveis.

Tipo: opcional.

Descrição Resumida

O Usuário pode atualizar a tela principal a qualquer momento e verificar os mapas disponíveis.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa.

Fluxo de Eventos

1 – Usuário informa requisição de atualização de tela ao servidor.

2 – Servidor retorna lista de mapas disponíveis ao cliente.

Fluxo Alternativo para o passo 1, caso ocorra algum problema na conexão durante a solicitação de atualização.

1.1 – Mensagem, “Falha na conexão com o servidor.”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso cliente não recebe nova lista do servidor.

2.1 – Mensagem, “Verifique a sua conexão com o servidor.”.

2.2 – Encerra caso de uso.

Cenário Principal

Silva deseja atualizar sua tela principal para verificar os mapas disponíveis para exploração, então ele clica no botão “Atualizar”, na barra de botões principais na parte inferior da tela.

Diagrama de Seqüência

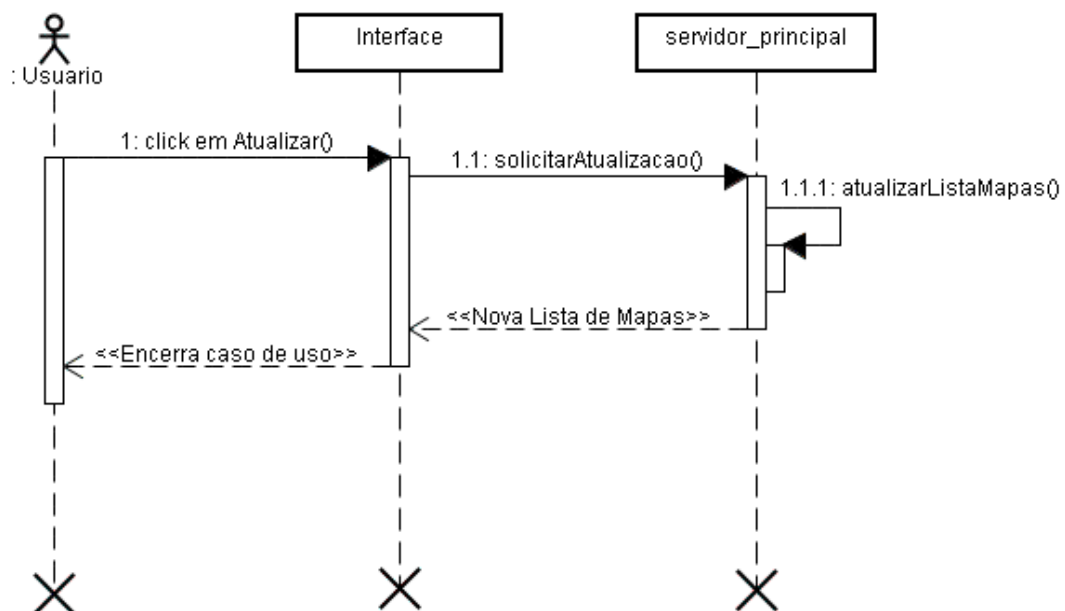


Figura 51 – Diagrama de Seqüência (Atualizar Lista de Mapas).

7.7.1.16 C.U. Limpar Conversa

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Usuário

Propósito: Usuário pode limpar sua tela de conversa.

Tipo: opcional.

Descrição Resumida

O Usuário pode limpar sua tela de conversa a qualquer momento.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa e com uma conversa estabelecida.

Fluxo de Eventos

- 1 – Usuário clica no botão “Limpar Conversa”.
- 2 – Sistema limpa conversa na tela de conversa.

Cenário Principal

Silva deseja limpar sua tela de conversa, pois sua esposa está ao seu lado, então ele clica no botão “Limpar Conversa” no canto superior esquerdo da tela de conversa e limpa sua tela.

Diagrama de Seqüência

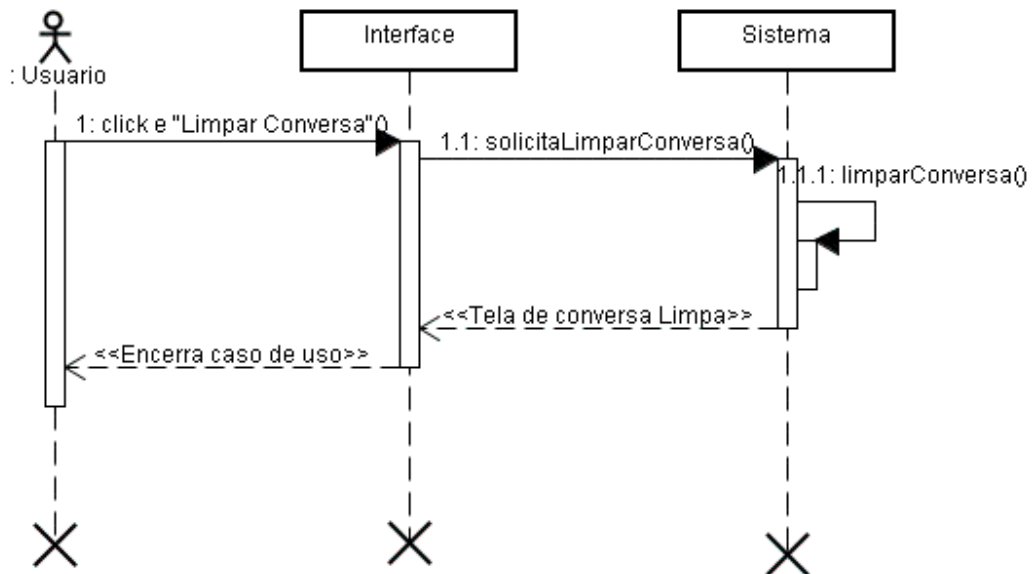


Figura 52 – Diagrama de Seqüência (Limpar Conversa).

7.7.1.17 C.U. Salvar Conversa

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Usuário

Propósito: Usuário pode salvar sua conversa com outro usuário.

Tipo: opcional.

Descrição Resumida

O Usuário pode salvar sua conversa com outro usuário a qualquer momento.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e possuir o programa instalado no computador do qual está efetuando o acesso, estar logado no programa e com uma conversa estabelecida.

Fluxo de Eventos

- 1 – Usuário clica no botão “Salvar Conversa”.
- 2 – Sistema salva conversa em um arquivo e em um diretório escolhido pelo usuário.

Cenário Principal

Silva deseja salvar conversa com uma amiga, então ele clica no botão “Salvar Conversa” no canto superior esquerdo da tela de conversa, escolhe o diretório onde deseja salvar e salva o conteúdo da conversa.

Diagrama de Seqüência

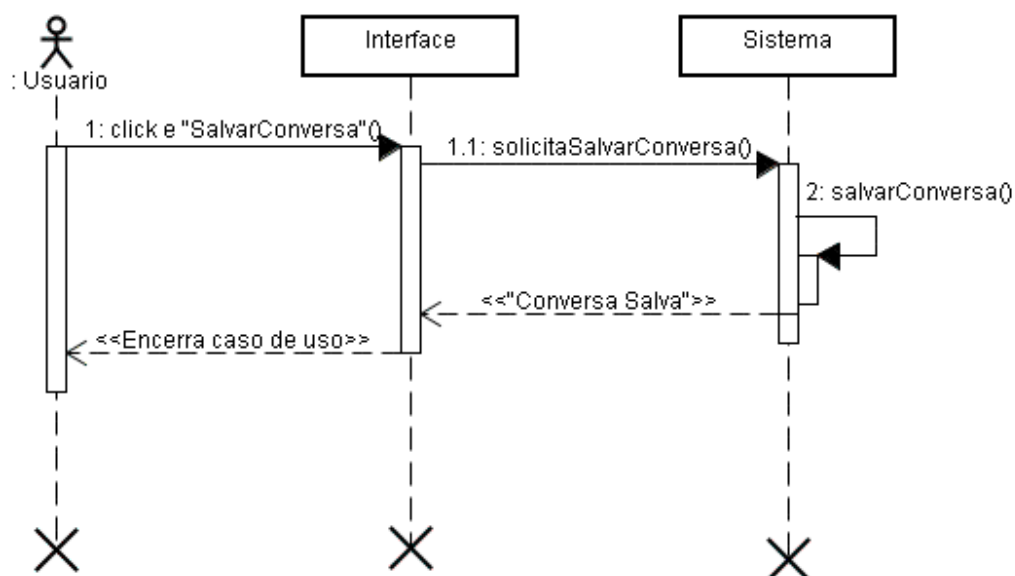


Figura 53 – Diagrama de Seqüência (Salvar Conversa).

7.7.1.18 C.U. Desconectar Usuário

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal, Administrador do Servidor de Mapas.

Propósito: O administrador pode desconectar um usuário do programa ou do mapa.

Tipo: opcional.

Descrição Resumida

O administrador pode desconectar um usuário conectado ao programa ou do mapa no momento em que achar conveniente.

Pré-Condições

Ser administrador dos sistemas, possuindo login nos aplicativos.

Fluxo de Eventos

1 – Administrador clica no botão “Desconectar Usuário”.

2 – Servidor desconecta usuário.

Fluxo alternativo para o passo 2, caso não haja nenhum usuário conectado ou selecionado.

2.1 – Mensagem “Selecione um usuário disponível”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja desconectar um certo usuário em um dos aplicativos dos servidores, então ele seleciona o usuário desejado na lista de usuário e clica em “Kickar Usário”, então o usuário escolhido é desconectado do servidor.

Diagrama de Seqüência

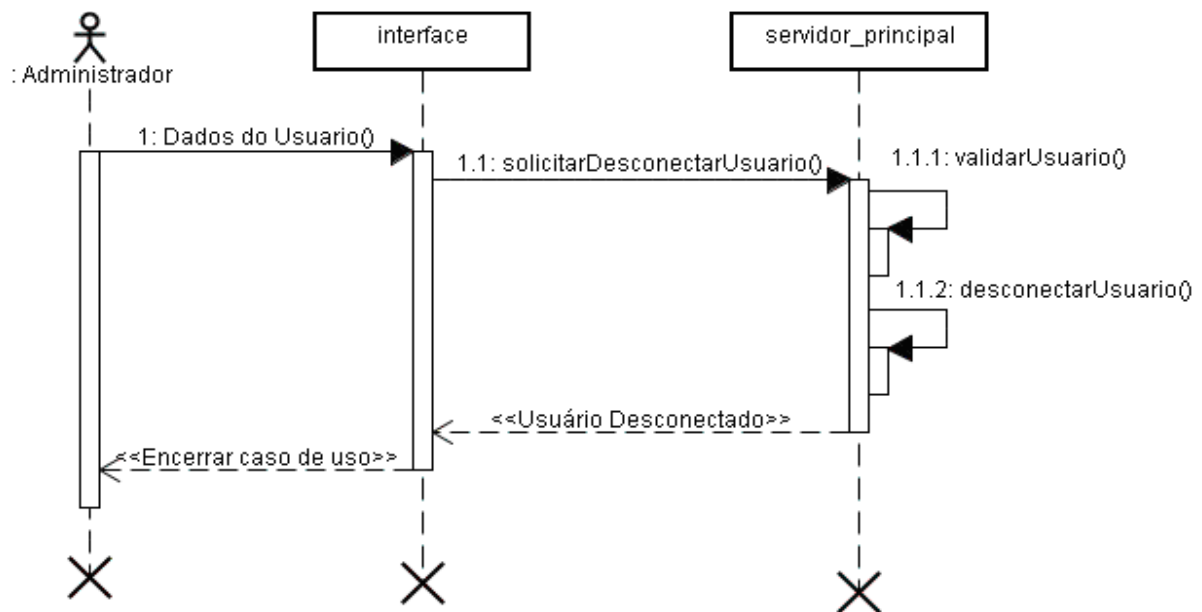


Figura 54 – Diagrama de Seqüência (Desconectar Usuário).

7.7.1.19 C.U. Desconectar Todos

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal, Administrador do Servidor de Mapas.

Propósito: O administrador pode desconectar todos usuários do programa ou do mapa.

Tipo: opcional.

Descrição Resumida

O administrador pode desconectar todos usuários conectados ao programa ou do mapa no momento em que achar conveniente.

Pré-Condições

Ser administrador do sistema, possuindo login nos aplicativos.

Fluxo de Eventos

1 – Administrador clica no botão “Desconectar Todos”.

2 – Servidor desconecta todos usuários.

Fluxo alternativo para o passo 2, caso não haja nenhum usuário conectado.

2.1 – Mensagem “Não há usuários conectados no momento”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja desconectar todos usuários em um dos aplicativos dos servidores, então ele clica em “Kickar Todos”, então o usuários são desconectados do servidor.

Diagrama de Seqüência

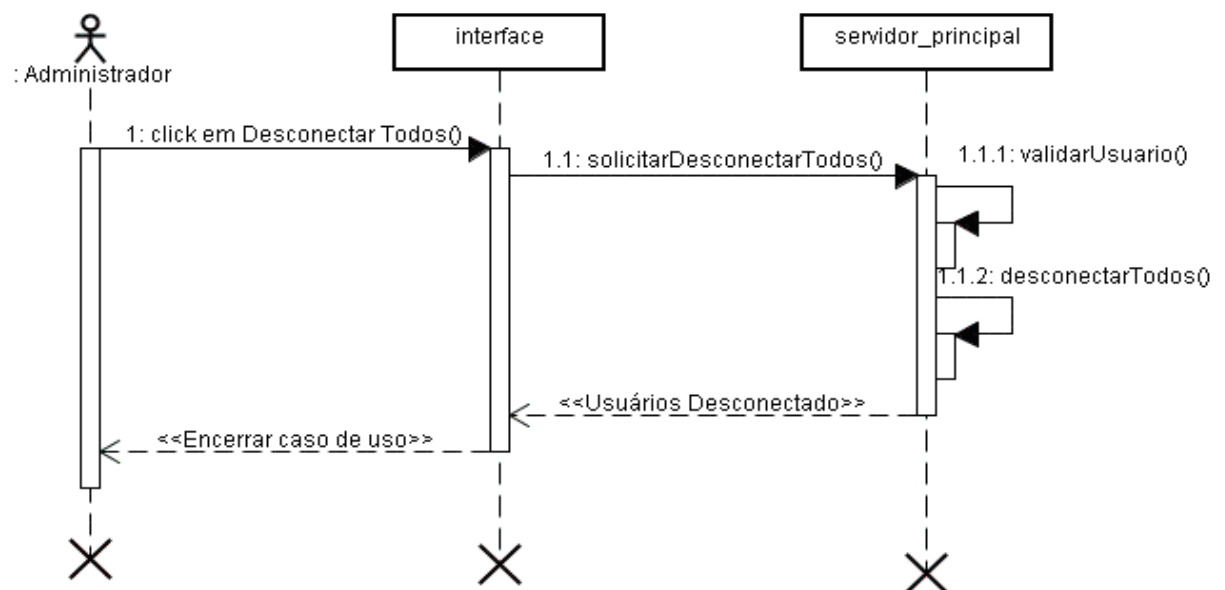


Figura 55 – Diagrama de Seqüência (Desconectar Todos os Usuário).

7.7.1.20 C.U. Cadastrar Modelo

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode cadastrar novos avatares.

Tipo: opcional.

Descrição Resumida

O administrador pode cadastrar novos modelos no banco de dados através do aplicativo do servidor principal.

Pré-Condições

Ser administrador do sistema, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

1 – Administrador envia dados do novo avatar ao servidor.

2 – Servidor cadastra avatar.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

1.1 – Mensagem “Preencha corretamente os campos!”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

2.1 – Mensagem “Avatar com estes dados já cadastrado.”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja cadastrar um novo modelo no banco de dados, então ele loga-se no programa do servidor principal, clica em “Avatares”, após clica em “Novo” preenche os dados no lado esquerdo, abre o arquivo do avatar desejado e confirma a operação.

Diagrama de Seqüência

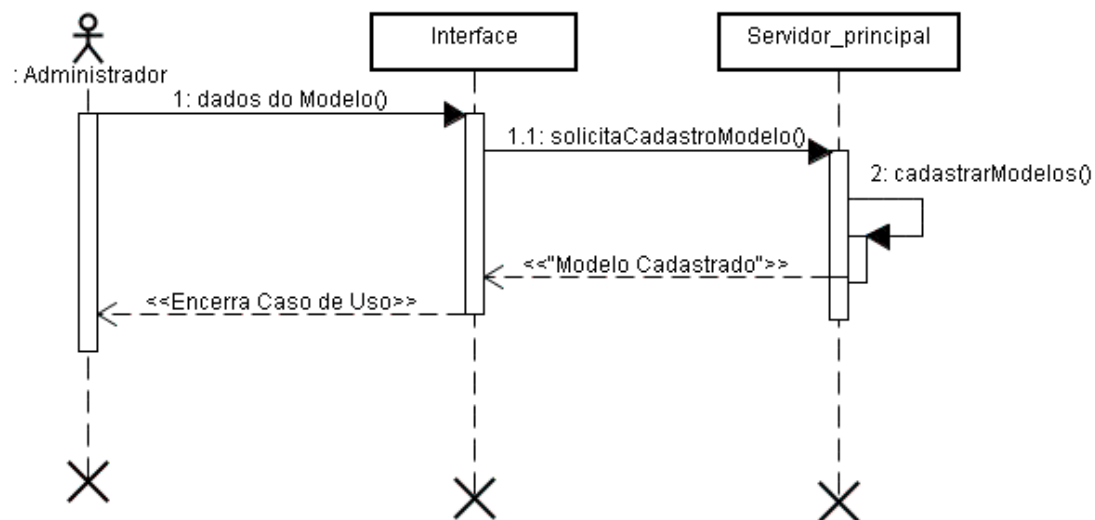


Figura 56 – Diagrama de Seqüência (Cadastrar Modelo).

7.7.1.21 C.U. Cadastrar Servidor de Mapas

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode cadastrar novos servidores de mapas.

Tipo: opcional.

Descrição Resumida

O administrador pode cadastrar novos servidores de mapas no banco de dados através do aplicativo do servidor principal, disponibilizando novos cenário para exploração 3D.

Pré-Condições

Ser administrador do sistemas, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

1 – Administrador envia dados do novo servidor de mapas ao servidor.

2 – Servidor cadastra servidor de mapas.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

1.1 – Mensagem “Preencha corretamente os campos!”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

2.1 – Mensagem “Servidor com estes dados já cadastrado.”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja cadastrar um novo servidor de mapa no banco de dados, então ele loga-se no programa do servidor principal, clica em “Servidores”, após clica em “Novo” preenche os dados no lado esquerdo, abre o arquivo do mapa desejado e confirma a operação.

Diagrama de Sequência

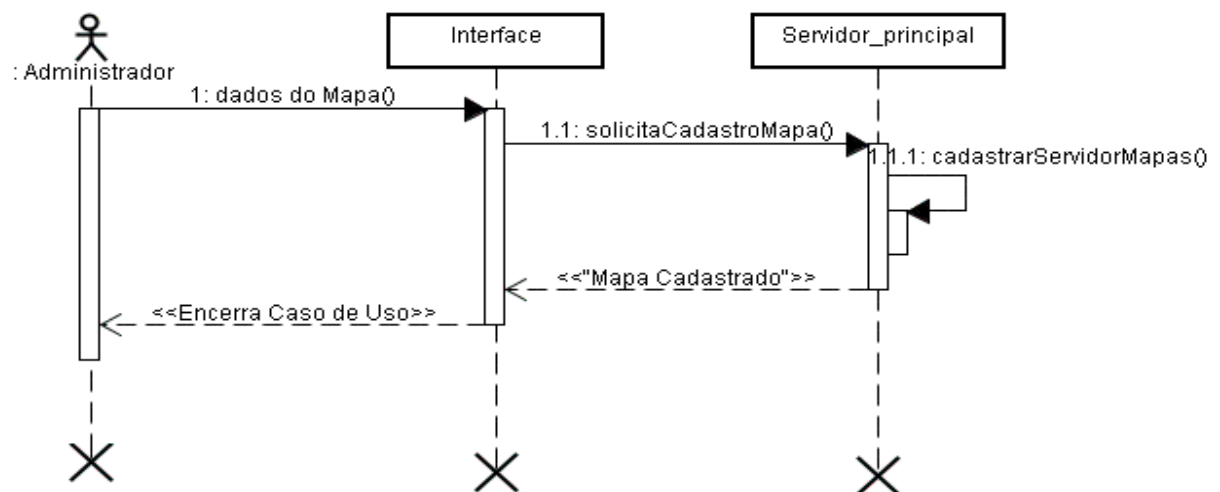


Figura 57 – Diagrama de Sequência (Cadastrar Servidor de Mapas).

7.7.1.22 C.U. Editar Servidor de Mapas

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode editar os servidores de mapas.

Tipo: opcional.

Descrição Resumida

O administrador pode editar os servidores de mapas existentes no banco de dados através do aplicativo do servidor principal.

Pré-Condições

Ser administrador do sistema, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

1 – Administrador escolhe servidor e envia dados editados ao servidor principal.

2 – Servidor atualiza os dados do servidor de mapas.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

1.1 – Mensagem “Preencha corretamente os campos!”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

2.1 – Mensagem “Servidor com estes já cadastrados.”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja editar um servidor de mapa no banco de dados, então ele loga-se no programa do servidor principal, clica em “Servidores”, seleciona o servidor desejado para alteração, após clica em “Editar” realiza as modificações e confirma a operação.

Diagrama de Seqüência

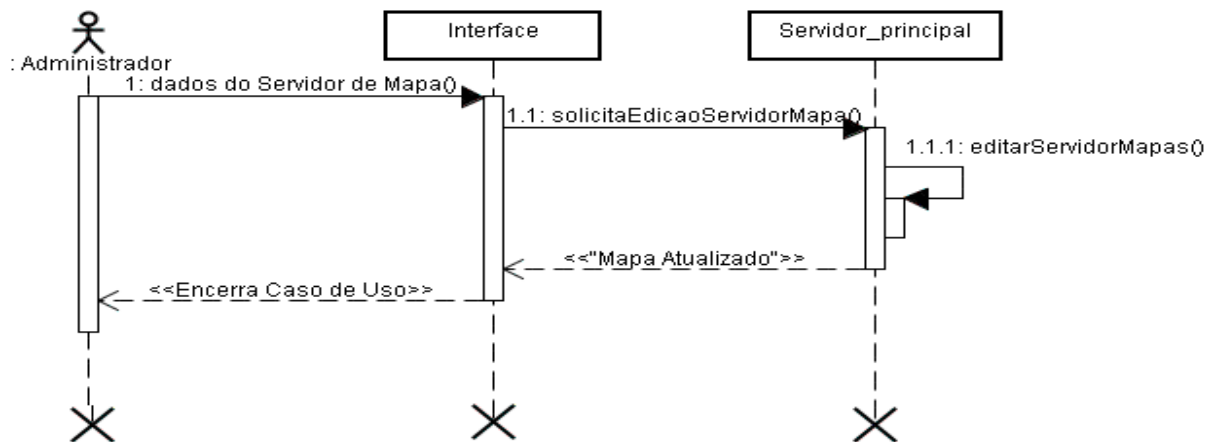


Figura 58 – Diagrama de Seqüência (Editar Servidor de Mapas).

7.7.1.23 C.U. Editar Modelos

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode editar os modelos.

Tipo: opcional.

Descrição Resumida

O administrador pode editar os avatares existentes no banco de dados através do aplicativo do servidor principal.

Pré-Condições

Ser administrador do sistemas, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

- 1 – Administrador escolhe um avatar e envia dados editados ao servidor principal.
- 2 – Servidor atualiza os dados do avatar.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

- 1.1 – Mensagem “Preencha corretamente os campos!”.
- 1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

- 2.1 – Mensagem “Servidor com estes dados já cadastrado.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja editar um modelo no banco de dados, então ele loga-se no programa do servidor principal, clica em “Avatares”, seleciona o modelo desejado para alteração, após clica em “Editar” realiza as modificações e confirma a operação.

Diagrama de Sequência

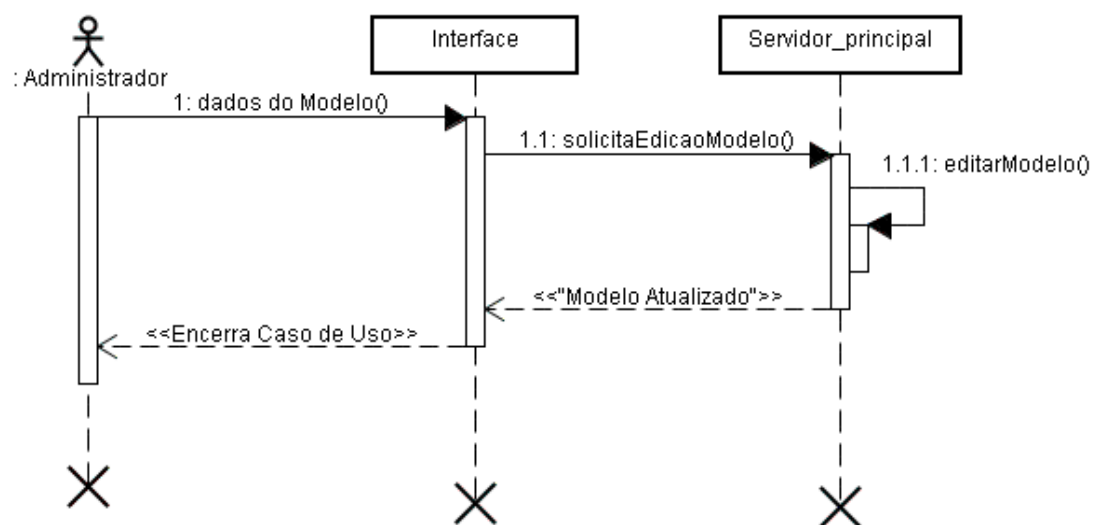


Figura 59 – Diagrama de Seqüência (Editar Modelo).

7.7.1.24 C.U. Excluir Servidor de Mapas

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode excluir os servidores de mapas.

Tipo: opcional.

Descrição Resumida

O administrador pode excluir os servidores de mapas existentes no banco de dados através do aplicativo do servidor principal.

Pré-Condições

Ser administrador do sistemas, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

1 – Administrador escolhe servidor e envia dados ao servidor principal.

2 – Servidor exclui os dados do servidor de mapas.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

1.1 – Mensagem “Preencha corretamente os campos!”.

1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

2.1 – Mensagem “Servidor com estes dados já cadastrado.”.

2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja excluir um servidor de mapa no banco de dados, então ele loga-se no programa do servidor principal, clica em “Servidores”, seleciona o servidor desejado para alteração, após clica em “Excluir” realiza as modificações e confirma a operação.

Diagrama de Seqüência

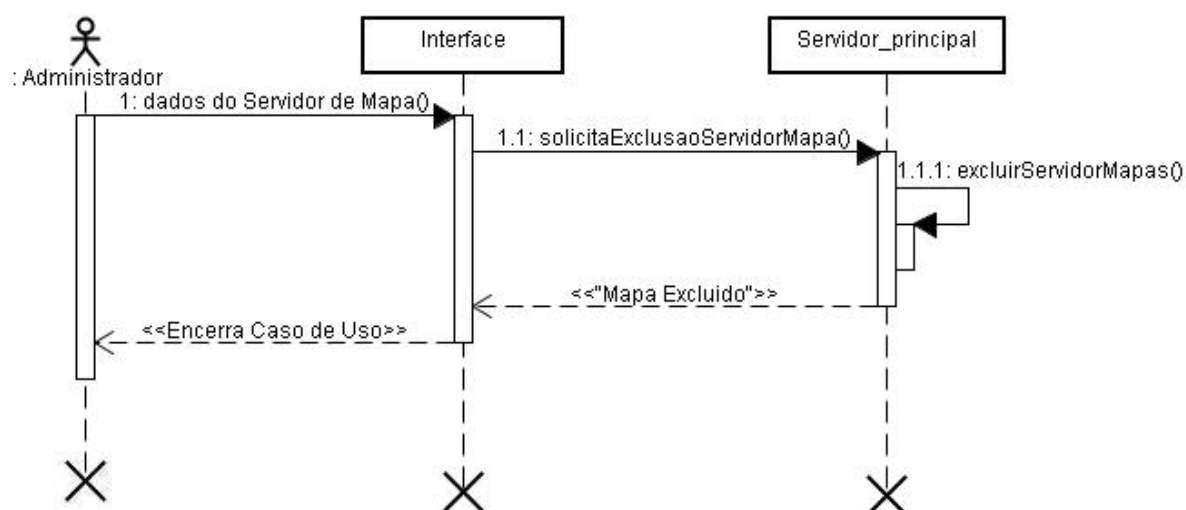


Figura 60 – Diagrama de Seqüência (Excluir Servidor de Mapa).

7.7.1.25 C.U. Excluir Modelos

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Gabriel

Ator: Administrador do Servidor Principal.

Propósito: O administrador pode excluir os modelos.

Tipo: opcional.

Descrição Resumida

O administrador pode excluir os avatares existentes no banco de dados através do aplicativo do servidor principal.

Pré-Condições

Ser administrador do sistemas, possuindo login no aplicativo do servidor principal.

Fluxo de Eventos

- 1 – Administrador escolhe um avatar e envia dados ao servidor principal.
- 2 – Servidor atualiza os dados do avatar.

Fluxo alternativo para o passo 1, caso algum campo seja preenchido de forma incorreta.

- 1.1 – Mensagem “Preencha corretamente os campos!”.
- 1.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 2, caso os dados escolhidos já estejam cadastrados.

- 2.1 – Mensagem “Servidor com estes dados já cadastrado.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

Norris deseja excluir um modelo no banco de dados, então ele loga-se no programa do servidor principal, clica em “Avatares”, seleciona o modelo desejado para alteração, após clica em “Excluir” realiza as modificações e confirma a operação.

Diagrama de Seqüência

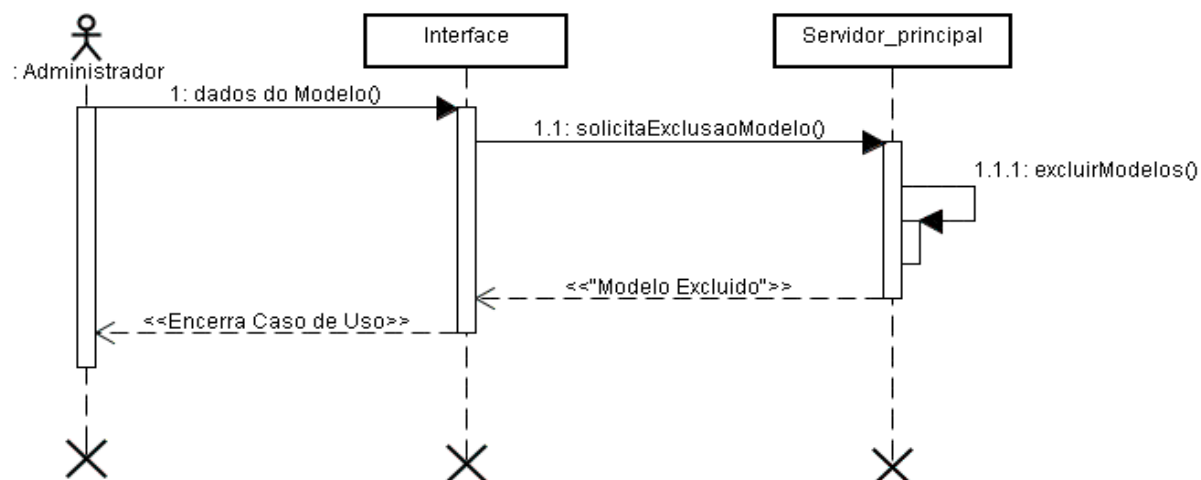


Figura 61 – Diagrama de Seqüência (Excluir Modelo).

7.7.1.26 C.U. Fazer Login

PRIORIZAÇÃO

Prioridade: Alta

Autor: Felipe

Ator: Servidor de Mapas

Propósito: Logar o administrador no aplicativo de servidor de mapas.

Tipo: indispensável.

Descrição Resumida

O administrador de mapas deve realizar o login na aplicação de servidor de mapas para ter acesso aos seus dados e ações.

Pré-Condições

Possuir login válido no aplicativo de administração de mapas.

Fluxo de Eventos

- 1 – Usuário informa dados para logar-se (login e senha).
- 2 – Chama caso de uso CONECTAR A UM SERVIDOR.
- 3 – Servidor Principal valida login.
- 4 – Servidor Principal loga usuário.

Fluxo Alternativo para o passo 3, caso login esteja incorreto.

- 2.1 – Mensagem, “Login Inválido.”.
- 2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 3, caso senha esteja incorreta.

- 2.1 – Mensagem, “Senha Inválida.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

O administrador do Servidor de Mapas deseja realizar o login na aplicação, para isso ele abre o arquivo executável do servidor de mapas, posteriormente insere seu login e sua senha, após está conectado ao servidor de mapas, podendo visualizar suas ações.

Diagrama de Seqüência

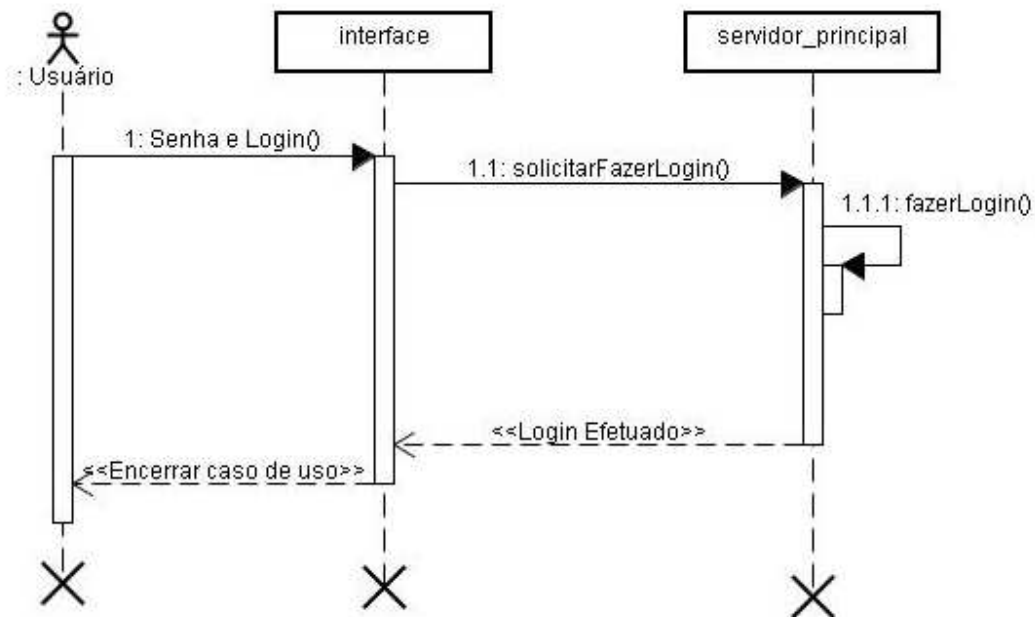


Figura 62 – Diagrama de Seqüência (Fazer Login no Servidor de Mapa).

7.7.1.27 Interpretar Mensagens

PRIORIZAÇÃO

Prioridade: Alta

Autor: Felipe

Ator: Servidor de Mapas, Servidor Principal

Propósito: Servidor recebe e interpreta mensagem de clientes.

Tipo: indispensável.

Descrição Resumida

O servidor de mapas pode receber e interpretar certas mensagens enviadas a ele pelo servidor principal e por clientes.

Pré-Condições

Os servidores estarem ligados em rede.

Fluxo de Eventos

- 1 – Servidor de Mapas recebe uma mensagem do servidor principal.
- 2 – Servidor de mapas interpreta mensagem.

Cenário Principal

O Servidor de Mapas recebe uma mensagem do servidor principal, ele interpreta e descobre que aquele está conectado a este.

Diagrama de Seqüência

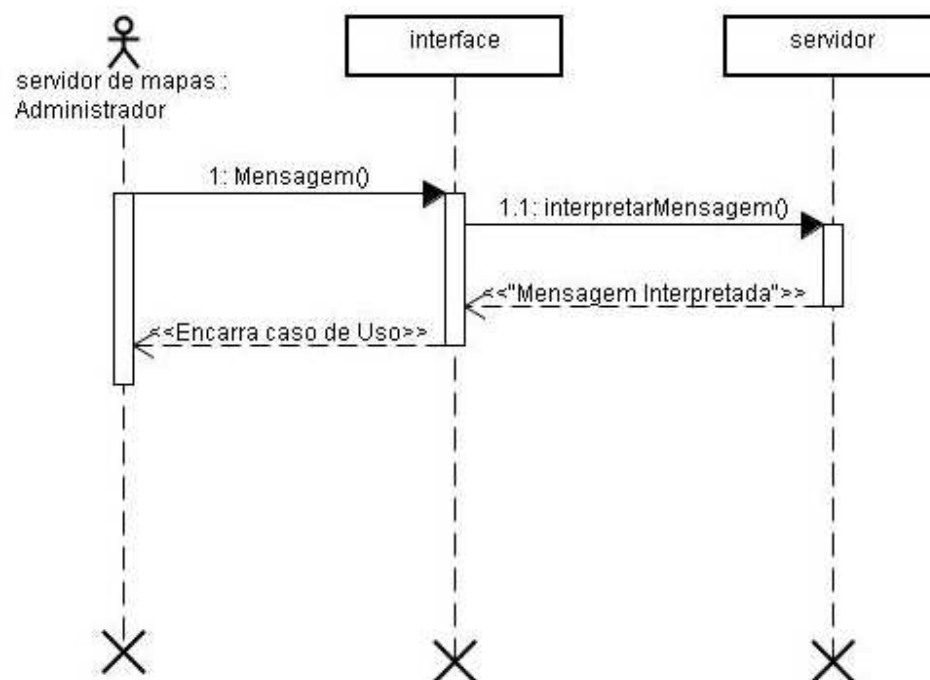


Figura 63 – Diagrama de Seqüência (Interpretar Mensagem).

7.7.2 Casos de Uso do Site

7.7.2.1 C.U. Fazer Login

PRIORIZAÇÃO

Prioridade: Alta

Autor: Felipe

Ator: Usuário

Propósito: Logar o usuário no sistema do site.

Tipo: indispensável.

Descrição Resumida

O usuário deve possuir seu cadastro para realizar o login no programa, e após o este realizado é feita a conexão com o servidor, para então usufruir das sessões de download do site.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH.

Fluxo de Eventos

- 1 – Usuário informa dados para logar-se (login e senha).
- 2 – Chama caso de uso CONECTAR A UM SERVIDOR.
- 3 – Servidor valida login.
- 4 – Servidor loga usuário.

Fluxo Alternativo para o passo 3, caso login esteja incorreto.

- 2.1 – Mensagem, “Login Inválido.”.
- 2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 3, caso senha esteja incorreta.

2.1 – Mensagem, “Senha Inválida.”.

2.2 – Encerra caso de uso.

Cenário Principal

O usuário Silva visita o site 3DBLAH e realiza seu cadastro, então com acesso a sessão de “Downloads” e realiza o seu login, se validado é direcionado a interface de escolha da sessão de downloads.

Diagrama de Seqüência

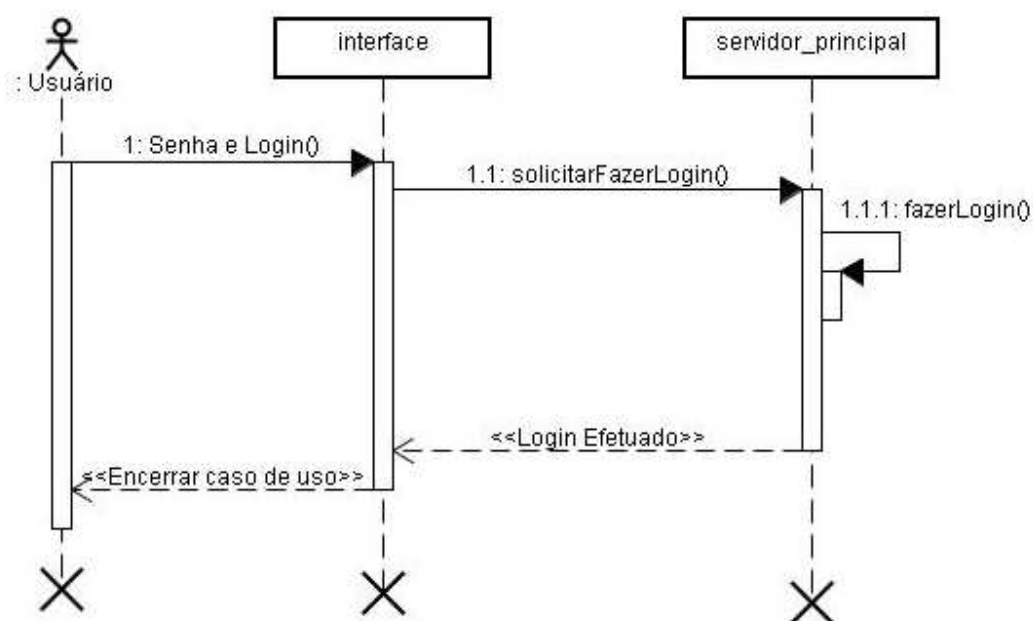


Figura 64 – Diagrama de Seqüência (Fazer Login no Site).

7.7.2.2 C.U. Fazer Logoff

PRIORIZAÇÃO

Prioridade: Média

Autor: Rafael

Ator: Usuário

Propósito: Efetuar logoff do usuário no servidor.

Tipo: indispensável.

Descrição Resumida

O usuário pode adicionar realizar o Logoff, saindo apenas de sua conta sem fechar a página da internet.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e estar logado no mesmo.

Fluxo de Eventos

- 1 – Usuário efetua logoff.
- 2 – Servidor envia mensagem para confirmação de logoff.
- 3 – Usuário confirma logoff.
- 4 – Servidor desloga usuário.

Fluxo Alternativo para o passo 3, caso usuário desista de deslogar-se.

- 3.1 – Encerra caso de uso.

Cenário Principal

O usuário Silva deseja agora realizar o log off de sua conta no site para entrar com outra conta, então ele clica no botão “Sair” ao lado de seu nome, no canto mediano

direito da tela e após a confirmação na mensagem enviada pelo servidor, ele é deslogado do sistema.

Diagrama de Seqüência

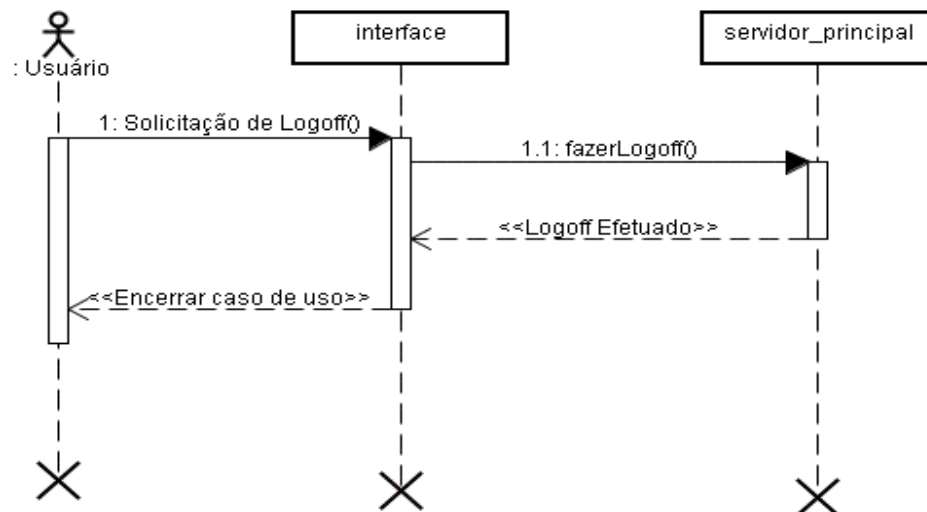


Figura 65 – Diagrama de Seqüência (Fazer Logoff do Usuário do Servidor).

7.7.2.3 C.U. Enviar Mensagem

PRIORIZAÇÃO

Prioridade: Alta

Autor: Rodrigo

Ator: Usuário

Propósito: Permitir o envio de mensagens ao suporte on-line do programa.

Tipo: indispensável.

Descrição Resumida

O usuário pode enviar mensagens de críticas, sugestões e dúvidas ao suporte do 3DBLAH.

Pré-Condições

Estar na página do programa, na sessão “Suporte”.

Fluxo de Eventos

- 1 – Usuário envia mensagem ao servidor.
- 2 – Servidor retorna confirmação de envio de mensagem ao usuário.

Fluxo Alternativo para o passo 2, caso o cliente não receba confirmação de envio.

- 2.1 – Mensagem, “Sua mensagem não pode ser enviada. Tente Novamente.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

Silva visita o site 3DBLAH e resolve enviar uma mensagem ao nosso suporte para saber quando estarão disponíveis mais modelos para escolha de avatar, então ele entra na sessão “Suporte”, preenche os campos que são obrigatórios e digita a mensagem, apo clica em “Enviar” e aguarda a resposta do suporte.

Diagrama de Seqüência

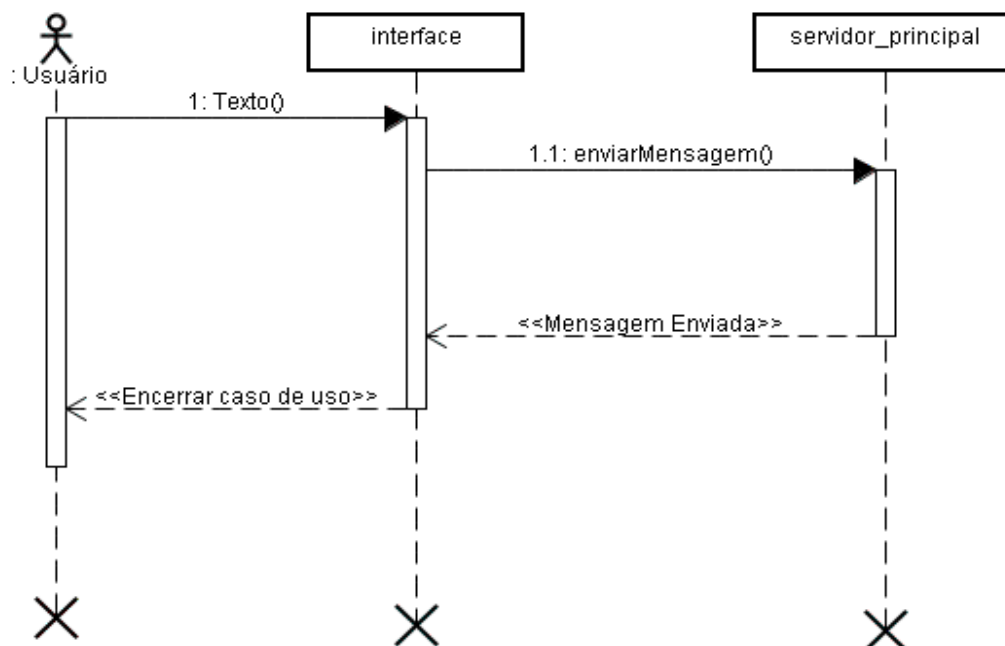


Figura 66 – Diagrama de Seqüência (Enviar Mensagem via Site).

7.7.2.4 C.U. Download

PRIORIZAÇÃO

Prioridade: Baixa

Autor: Rodrigo

Ator: Usuário

Propósito: Permitir download de arquivos para os usuários.

Tipo: opcional.

Descrição Resumida

O usuário pode realizar o download do programa, modelos, mapas e atualizações a partir do site.

Pré-Condições

Ter um cadastro realizado no site 3DBLAH e estar logado no mesmo.

Fluxo de Eventos

- 1 – Usuário entra na sessão escolhida e solicita download.
- 2 – Servidor verifica disponibilidade do arquivo.
- 3 – Servidor direciona o arquivo que está em um servidor ftp para o usuário.

Fluxo Alternativo para o passo 2, caso o arquivo esteja indisponível.

- 2.1 – Mensagem, “Arquivo indisponível para download.”.
- 2.2 – Encerra caso de uso.

Cenário Principal

Silva entra no site, vai para a sessão de *download*, insere seus dados e realiza o login. Após decide fazer o download de um novo cenário disponível, então ele entra na sessão “Mapas” e após clica no botão “Download” localizado ao lado do mapa desejado, por fim ele seleciona o diretório de “Maps” que está dentro do diretório no qual o jogo foi instalado.

Diagrama de Seqüência

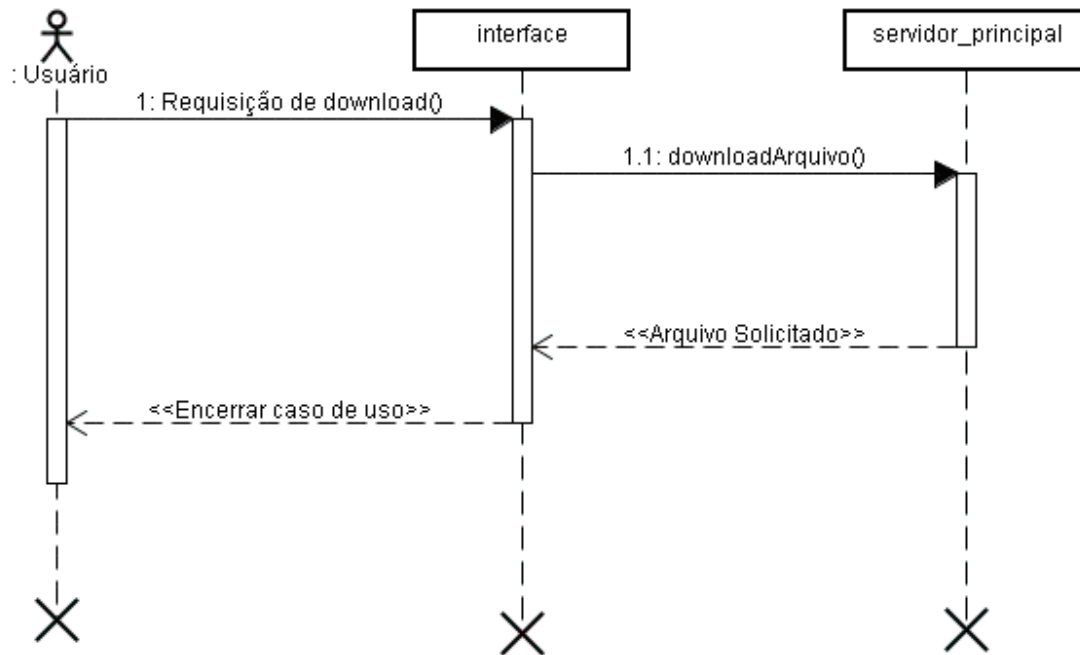


Figura 67 – Diagrama de Seqüência (*Download* de Arquivos do Site).

7.7.2.7 C.U. Cadastrar Usuário

PRIORIZAÇÃO

Prioridade: Alta

Autor: Gabriel

Ator: Usuário

Propósito: Cadastrar usuário, ponto fundamental para utilização do programa.

Tipo: indispensável.

Descrição Resumida

O usuário deve cadastrar-se no site do programa para obter sua senha e login, e poder executar o programa e realizar downloads no site.

Pré-Condições

O usuário acessar o site, o site estar disponível e entrar na sessão de cadastro.

Fluxo de Eventos

- 1 – Usuário acessa o site do programa e entra em ‘Cadastrar Novo Usuário’.
- 2 – Dados do cadastro são enviados ao servidor.
- 3 – Servidor valida dados do usuário.
- 4 – Servidor cadastra usuário na classe usuário.
- 5 – Servidor retorna mensagem de confirmação de cadastro.

Fluxo Alternativo para o passo 2, caso ocorra algum problema na conexão durante a transferência dos dados.

- 2.1 – Mensagem, “Os dados referentes ao cadastro não puderam ser enviados ao servidor.”.
- 2.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 3, caso os dados estejam preenchidos de maneira incorreta.

- 3.1 – Mensagem, “Dados Inválidos.”.
- 3.2 – Encerra caso de uso.

Fluxo Alternativo para o passo 4, caso seja impossível no momento cadastrar o usuário.

- 4.1 – Mensagem, “Ocorreu um erro durante o processo de cadastro de usuário.”.
- 4.2 – Encerra caso de uso.

Cenário Principal

Silva acessa o site 3DBLAH, conhece o programa e entra na sessão de cadastro. Ele preenche os campos necessários e envia seus dados ao servidor, o qual cadastra os dados do usuário no banco de dados, então o servidor envia a mensagem de confirmação de cadastro ao usuário.

Diagrama de Seqüência

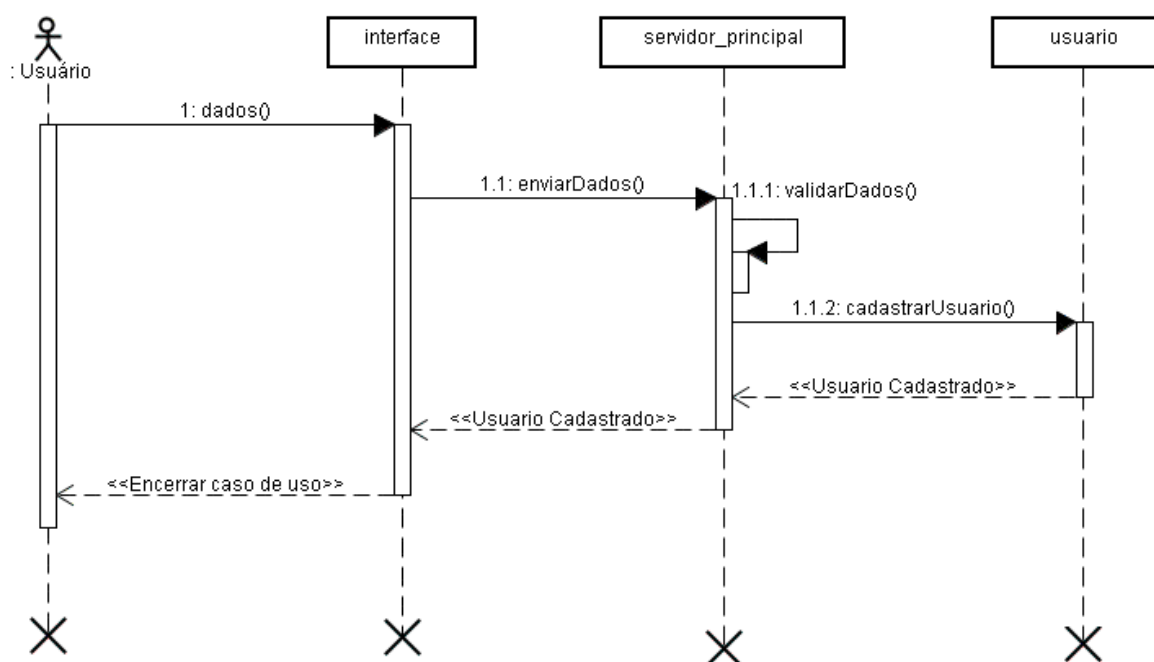


Figura 68 – Diagrama de Seqüência (Cadastro do Usuário no Site).

8 CONCLUSÃO

Neste projeto houve uma inovação nos aplicativos chamados Instant Messengers, com uma proposta de uma ambientação 3D, na qual é possível interagir com o cenário e com os outros modelos 3D, possuindo ainda as funcionalidades comuns dos softwares de mesma categoria, como troca de mensagens e arquivos.

Nele é possível reproduzir cenários comuns ao dia-a-dia ou mesmos pontos turísticos virtualmente, podendo até realizar o chamado “turismo virtual”, além de entreter o usuário com as possibilidades de mudar seu personagem e marcar encontros com pessoas e fazer novas amizades.

Acreditamos que nosso objetivo geral e específico foi alcançado com sucesso, pela praticidade do programa, as funcionalidades implementadas e a facilidade de uso e velocidade no tráfego de dados.

9 REQUISITOS MÍNIMOS PARA O SISTEMA

9.1 Para o Servidor

- CPU 3.0 GHz;
- 1 Gb de memória RAM;
- Internet 4 mbps ou mais;
- Porta 2255 disponível;
- Para o servidor de mapas, porta 2256 disponível.

9.2 Para o Cliente

- CPU 500MHz;
- 256 MB de memória RAM ;
- Placa de vídeo 3D (Recomendável);
- Acesso à Internet banda larga;
- Porta 2255 disponível;
- Porta 2256 disponível.

10 DIFICULDADES ENCONTRADAS

Durante o desenvolvimento do projeto surgiram algumas dificuldades, como alguns bugs que aparecem quando o usuário entra pela segunda vez em alguma mapa. Este bug estáá relacionados à engine utilizada.

Porém nossa maior dificuldade foi em conseguir um servidor para a aplicação, o que ficou a cargo da coordenação do curso de Tecnologia em Informática da Universidade Federal do Paraná, setor Escola Técnica, sendo liberado apenas em fevereiro do ano de 2007. Após uma espera de aproximadamente 3 meses, decidimos conseguir algum servidor, e devido a isso perdemos algo em relação a capacidade e velocidade de processamento de dados.

A direção da Escola Técnica nos forneceu na data de 15 de janeiro de 2007, aproximadamente, um computado para utilizarmos como o servidor.

11 PROJETOS FUTUROS

Planejamos futuramente ampliar o mundo 3DBLAH, com constantes inovações e atualizações para entreter cada vez mais os usuários, como a possibilidade de jogos em 3D dentro do próprio aplicativo, como por exemplo você poder correr de kart com as pessoas presentes no seu mapa. Jogos mais simples como damas e jogo da velha, também estão em nossos planos.

Outro fator importante que queremos aliar ao 3DBLAH é proporcionar o uso da webcam no aplicativo com a possibilidade de salvar um pequeno trecho de vídeo no computador do usuário.

Achamos interessante também a implantação futura, na próxima versão do programa a possibilidade do usuário poder montar seu próprio personagem, modificando a cabeça, tronco e membros, para que haja uma identidade maior no avatar pessoal, juntamente com a possibilidades de cada modelo tridimensional possuir uma animação própria.

Mas nosso maior objetivo é realmente conseguir um patrocínio para desenvolver pontos turísticos virtuais, tornando o 3DBLAH uma ferramenta completa de Instant Messenger e de turismo virtual, sendo possível até acessar sites e realizar reservas em hotéis com o auxílio do programa.

12 Ambiente de Desenvolvimento

Processador	Memória RAM	HD
AMD Sempron 2.8 GHz	512 Mb	120 Gb
AMD Athlon 1.66 GHz	1 Gb	80 Gb
AMD Sempron 2.2 GHz	1Gb	2 x 80 Gb
Intel Pentium 4 3.2 GHz	1 Gb	120 Gb
AMD Sempron 3.7 GHz	1 Gb	120 Gb

*A internet utilizada possui 400 Kbps.

Tabela 1 – Hardware Utilizado

13 Bibliografia

About Delphi. Disponível em: <http://www.delphi.com/about/> Acesso em: 05 de fev. 2007.

About Truevision3D. Disponível em:
<http://www.truevision3d.com/about.php?sid=dba636a1ab313a9eb37a7949f9af722a> Acesso em: 05 de fev. 2007.

ANQUETIL, Nicolas. **Diagrama de atividade, diagrama de estado.** Disponível em:
<http://www.ucb.br/ucbtic/mgcti/paginapessoalprof/Nicolas/Disciplinas/UML/node7.html> Acesso em: 01 de mar. 2007.

Apostila Delphi sobre Sistemas Cliente Servidor. Disponível em:
<http://www.guiadodelphi.com.br/ler.php?codigo=1000> Acesso em: 10 de jan. 2007.

AZEVEDO, Eduardo. **Desenvolvimento de Jogos 3D e Aplicações em Realidade Virtual.** Rio de Janeiro: Elseiver, 2005.

CAMPOS, André. **573 - Desvendando a programação "Client/Server.** Arquivo de ajuda do Borland Delphi 7.

GUIDE, PMBOK. **A Guide to the Project Management Body of Knowledge.** Pennsylvania, USA: Project Management Institute Inc, 2004.

MilkShape 3D 1.7.10. Disponível em:
<http://baixaki.ig.com.br/download/MilkShape-3D.htm> Acesso em: 05 de fev. 2007.

PINHEIRO, Nilton. **Conhecendo o SQL Server 2005 Express Edition.** Disponível em: http://www.linhadecodigo.com.br/artigos.asp?id_ac=947&pag=1 Acesso em: 05 de fev. 2007.

QUADROS, M. **Gerência de Projetos de Software: técnicas e ferramentas.** Florianópolis: Visual Books, 2002.

VALERIANO, D. **Moderno gerenciamento de projetos.** São Paulo: Pearson Education do Brasil, 2005.

ANEXO I
PLANO DE PROJETO



UNIVERSIDADE FEDERAL DO PARANÁ

TECNOLOGIA EM INFORMÁTICA - 3º Ano - Tarde

Disciplina: Projetos

Projeto: 3DBLAH

Orientador: Roberto Raittz

3DBLAH

PLANO GERAL DE PROJETO

Equipe:

Alexandre Perin Marciano de Souza

Felipe Latif Mechailech

Gabriel Carvalho da Silva Neves

Rafael Scussel Michelotto

Rodrigo Cesar da Silva

3DBLAH
PLANO GERAL DE PROJETO

*Documento referente ao projeto de graduação
do curso de Tecnologia em Informática da
UFPR, intitulado “3DBLAH”.
Orientador: Roberto Raittz*

CURITIBA
2006

RESUMO

Este projeto orientado pelo professor Roberto Raittz, tem por finalidade criar um Instant Messenger um pouco diferente, em que além de conversar e conhecer novas pessoas é possível selecionar um personagem (avatar) e interagir livremente em diversos ambientes em 3D. Para realizar qualquer ação dentro do sistema o usuário deve estar logado e conectado ao servidor principal. As ações referentes ao ambiente 3D são gerenciadas por um outro servidor menor, que também deve estar logado e conectado ao servidor principal.

ABSTRACT

This project guided for the professor Roberto Tadeu Raittz, has for purpose to create an Instant Messenger a little different, where beyond talking and knowing new people it is possible to choose a character (an avatar) and to interact freely in diverse 3D environments. To carry through any action inside of the system the user must be log in and connected to the main server. The referring actions to the 3D environment are managed by one another lesser server, who also must be log in and connected to the main server.

I. INTRODUÇÃO

1. ESCOPO E PROPÓSITO DO DOCUMENTO

1.1 Instant Messenger

Ele é um programa que além de enviar as mensagens, armazena os contatos do usuário, o perfil editado pelo mesmo, e configurações especiais do usuário, sendo possível até o envio de arquivos, desde que o remetente e o destinatário encontrem-se ambos conectados na internet e com o programa conectado ao servidor e logado no mesmo.

1.2 Propósito do documento

Este documento serve como base para o desenvolvimento do aplicativo 3DBLAH, no qual encontraram-se descritas todas as funcionalidades, objetivos e metodologias usadas no sistema. Este projeto está destinado ao Trabalho de Conclusão de Curso para a graduação em Tecnologia em Informática da Universidade Federal do Paraná sob orientação do professor Roberto Tadeu Raitz.

1.3 Escopo

Criar um Instant Messenger, que tem como propósito uma comunicação em tempo real através de mensagens, enviadas ao contato de destino no mesmo momento em que fora solicitada. O sistema a ser desenvolvido disponibilizará a seus usuários além do serviço de conversa em tempo real, a possibilidade de cada contato possuir um personagem 3D no mundo virtual de 3DBLAH, o qual poderá interagir com um cenário recriado tridimensionalmente, explorando-o e interagindo com outros usuários, conhecendo novas pessoas e novos locais. Além de viajar virtualmente ainda poderão manter contatos e amizades no programa, através de uma lista de contatos pessoal, onde poderão ser visualizados o perfil de seus amigos e enviar mensagens, arquivos e animações, bem como poderão ser utilizados outras funcionalidades comuns em Instant Messengers.

2. OBJETIVOS DO PROJETO

2.1. Objetivos

Desenvolver um sistema que permita ao usuário conhecer lugares virtualmente, e ainda poder visitar com amigos, e até mesmo conhecer pessoas novas, um novo conceito de bate-papo. O Objetivo principal é a diversão e entretenimento do usuário aumenta sua lista de amigos e seu conhecimento, através do “turismo” e a história do cenário onde se encontra.

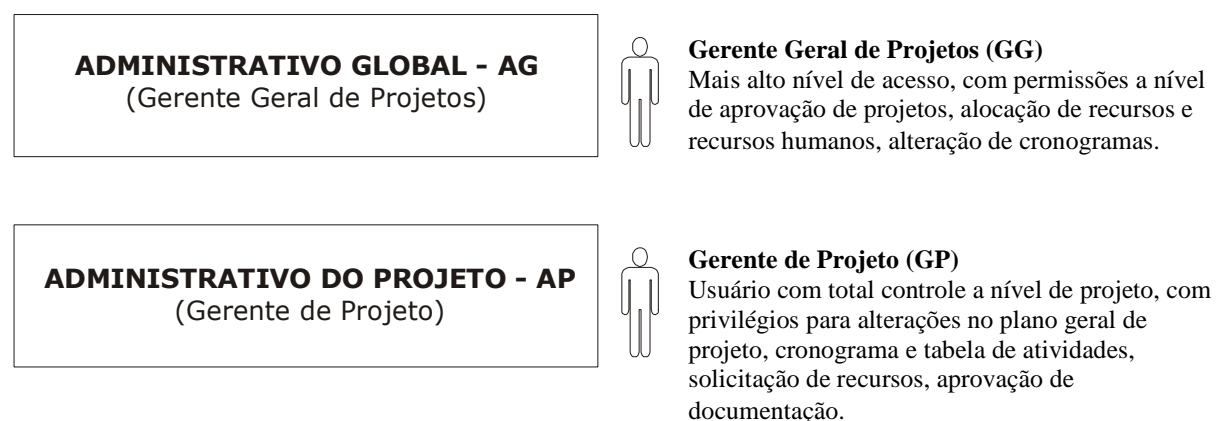
2.2. Funções principais

As funcionalidades deste sistema advêm da concepção de um bate-papo atual, como o MSN da Microsoft. O programa visa a interação do usuário com seus amigos e o cenário escolhido, e ainda desfrutando das vantagens e funcionalidades, das quais poderá desfrutar ao utilizar o programa 3DBLAH.

As funcionalidades presentes no programa, serão a criação de um personagem, o qual o usuário movimentará pelo cenário; a adição de contatos e à sua lista de amigos; possibilidade de edição do seu perfil e visualização do perfil de outros; possibilidade de envio de arquivos e mensagens; personalização de status (ausente, ocupado, dormindo, etc.); cadastro no site, necessário para login no servidor; o usuário contará com um suporte, através do nosso site.

2.3. Visão Geral

O sistema se divide em 3 módulos, como no modelo: (com seus respectivos atores)



ACOMPANHAMENTO - AC
(Membros da Equipe)



Membro da Equipe (ME)

Usuário com privilégios apenas de leitura, possibilitando o acompanhamento de funcionalidades que estão autorizadas a serem exibidas para este nível de acesso. Este ator pode ainda enviar relatórios e documentação.

2.4. Questões de desempenho

O desempenho da aplicação, a nível de velocidade dependerá do hardware e da conexão de internet do computador onde esta será hospedado o servidor do programa. Por ser uma aplicação voltada para o esquema cliente -servidor, a velocidade de conexão e o hardware do usuário também implica em seu desempenho usando o programa.

II. ESTIMATIVAS DE PROJETO

1. DADOS HISTÓRICOS USADOS NAS ESTIMATIVAS

A aplicação a ser desenvolvida tem conteúdo escasso para pesquisa a nível de software já desenvolvido, afinal a proposta do projeto é totalmente inovadora, não possuindo modelos e nem bibliografias a respeito. Embasar-nos-emos em programas já existentes para suporte, unindo livros sobre animação gráfica, técnica de desenvolvimento de jogos, e bate-papos em tempo real.

2. TÉCNICAS DE ETIMATIVAS

Para as estimativas foram utilizadas as seguintes técnicas:

2.1. Work Breakdown Structure (WBS)

Técnica que consiste em decompor etapas por nível hierárquico dentro de um projeto, resultando assim uma representação em forma de diagrama que ajuda a entender processos complexos, prazos e custos.

2.2. Project Evaluation and Review Technique (PERT)

Utilizado para mostrar relacionamentos e dependências entre as tarefas do projeto. O PERT usa cálculos estatísticos simples, de modo a calcular probabilidades para as datas de término das tarefas do projeto.

III. RISCOS DO PROJETO

1. ANÁLISE DE RISCOS

Principais riscos no desenvolvimento do projeto:

1.1. Identificação e Avaliação

- a.** Dificuldades enfrentadas com a linguagem de programação ou ferramentas a serem utilizadas

Avaliação: Médio

- b.** Construir uma modelagem de dados que não atenda as necessidades

Avaliação: Grave

- c.** Não cumprir o cronograma, ou não realizar tarefas no tempo especificado.

Avaliação: Grave

- d.** Perda ou extravio de documentos ou material do projeto

Avaliação: Grave

- e.** Indisponibilidade do sistema

Avaliação: Grave

2. ADMINISTRAÇÃO DOS RISCOS

2.1. Opções para evitar os riscos

- a.** Utilizar-se de referências bibliográficas, artigos, publicações ou apostilas para um

estudo aprofundado ou para encontrar respostas para as dúvidas encontradas. Também usar a comunicação entre membros da equipe ou com pessoas que tenham experiência.

- b.** Uma modelagem mal feita pode ocasionar a inconsistência dos dados. É necessário

então, o uso de ferramentas para modelagem, a constante revisão e avaliação do modelo de dados para verificar se as necessidades estão sendo atendidas.

- c. Dividir corretamente as tarefas, e fazendo com que todos façam de tudo um pouco, de modo a melhorar o desempenho e o conhecimento da equipe em si. Fazer verificações periódicas sobre o andamento das atividades e reuniões com a equipe e orientador.
- d. Criar mecanismos de padronização de documentos e de sua armazenagem em disco. Manter diretórios e arquivos de modo que os integrantes tenham a mesma visão sobre os mesmos. Definir uma política de backup a ser utilizada de forma periódica.
- e. De acordo com os recursos disponíveis, prover a redundância no ativo servidores e equipamentos que possam favorecer a disponibilidade do sistema.

2.2. Procedimentos de monitoração dos riscos

Realizar reuniões e avaliações periódicas com equipe e orientador de forma a identificar possíveis riscos que venham a ocorrer durante o projeto.

IV. CRONOGRAMA

Abaixo a descrição do cronograma, obtido com uso das técnicas de estimativas.

1. FASES DO PROJETO, SUAS ATIVIDADES E ESTIMATIVAS

1.1. Planejamento

a. Levantamento de Requisitos

Data de Início: 15/03/06 Data de Término: 31/03/06

b. Apresentação do Plano Geral de Projeto

Data de Início: 01/04/06 Data de Término: 30/04/06

1.2. Documentação

a. Diagrama de Casos de Uso

Data de Início: 01/05/06 Data de Término: 07/05/06

b. Diagrama de Classes

Data de Início: 08/05/06 Data de Término: 15/05/06

c. Diagrama de Sequência

Data de Início: 16/05/06 Data de Término: 23/05/06

d. Modelo Entidade-Relacionamento

Data de Início: 24/05/06 Data de Término: 31/05/06

e. Diagrama de Estado

Data de Início: 15/01/07 Data de Término: 01/02/07

f. Desenvolvimento do Relatório Final

Data de Início: 01/06/06 Data de Término: 05/02/07

1.3. Desenvolvimento

a. Modelo lógico do banco de dados

Data de Início: 01/06/06 Data de Término: 15/06/06

b. Módulo Administrativo Global

Data de Início: 16/06/06 Data de Término: 15/08/06

c. Módulo Administrativo do Projeto

Data de Início: 16/08/06 Data de Término: 16/09/06

d. Módulo Acompanhamento

Data de Início: 17/09/06 Data de Término: 31/10/06

1.4. Testes

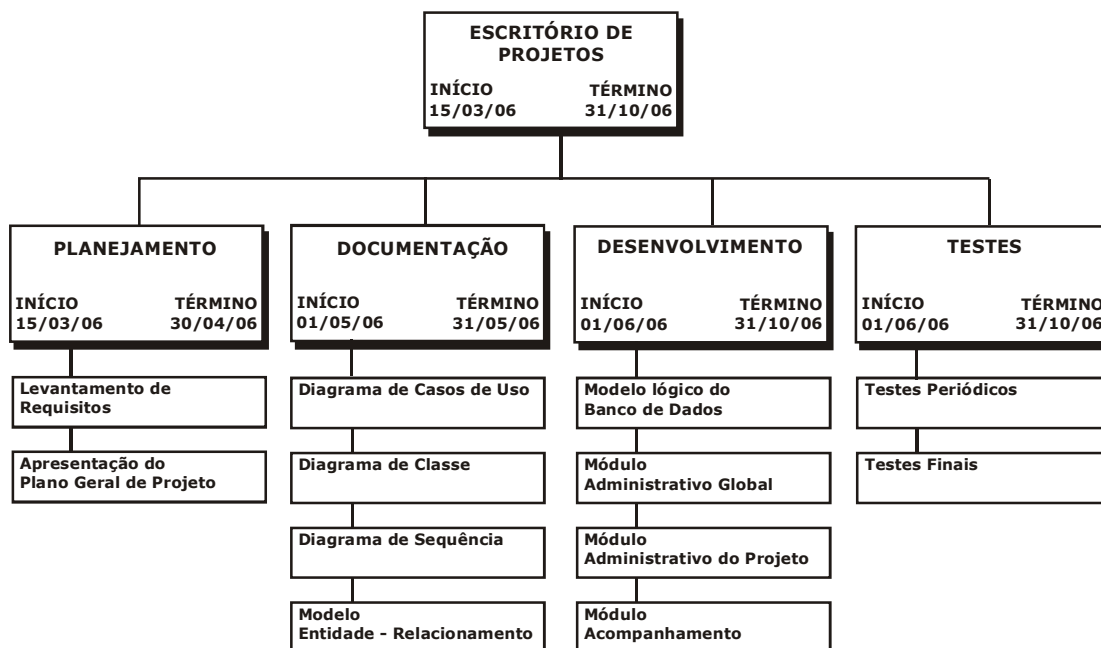
a. Testes Periódicos

Data de Início: 01/06/06 Data de Término: 31/10/06

b. Testes Finais

Data de Início: 01/10/06 Data de Término: 31/10/06

2. WORK BREAKDOWN STRUCTURE



V. RECURSOS DO PROJETO

1. PESSOAL

A equipe para desenvolver este projeto é formada por 5 alunos do curso de Tecnologia em Informática da Universidade Federal do Paraná, envolvidos nas fases de definição, execução e conclusão do projeto. Seguem os membros:

1. Alexandre Perin Marciano de Souza
2. Felipe Latif Mechailech
3. Gabriel Carvalho da Silva Neves
4. Rafael Scussel Michelotto
5. Rodrigo César da Silva

A equipe tem como orientador o professor Roberto Raitz.

2. HARDWARE

Infra estrutura da Escola Técnica da UFPR, rede, internet, computadores e servidores. Além destes equipamentos, usaremos computadores pessoais.

3. SOFTWARE

Abaixo o nome do software, sua versão e fornecedor:

3.1. Sistema Operacional

Windows XP SP2

Microsoft

3.2. Ferramenta CASE

JUDE Community

Eiwa System Management

3.3. Ferramentas para Banco de Dados

MS SQL

Microsoft

3.4. Ambientes de Desenvolvimento

Borland Delphi 7.0

Hammer

Macromedia Flash 8.0

Adobe Photoshop

Corel Draw X3

3.5 Navegadores

Mozilla Firefox 1.5

Internet Explorer 6.0

VI. ORGANIZAÇÃO DO PESSOAL

1. ESTRUTURA DA EQUIPE

Ver em **V. RECURSOS DO PROJETO**, *item PESSOAL*.

VII. CONSIDERAÇÕES FINAIS

Este Plano Geral de Projeto será submetido a avaliação pelo orientador responsável e será disponibilizado ao corpo de orientadores de projetos.

VIII. BIBLIOGRAFIA

GUIDE, PMBOK. A Guide to the Project Management Body of Knowledge. Pennsylvania, USA, Project Management Institute Inc, 2004.

QUADROS, M. Gerência de Projetos de Software: técnicas e ferramentas. Florianópolis, Visual Books, 2002.

VALERIANO, D. Moderno gerenciamento de projetos. Pearson Education do Brasil, 2005. p. 99-109

ANEXO II
TABELA DE PROTOCOLOS

TABELA DE PROTOCOLOS

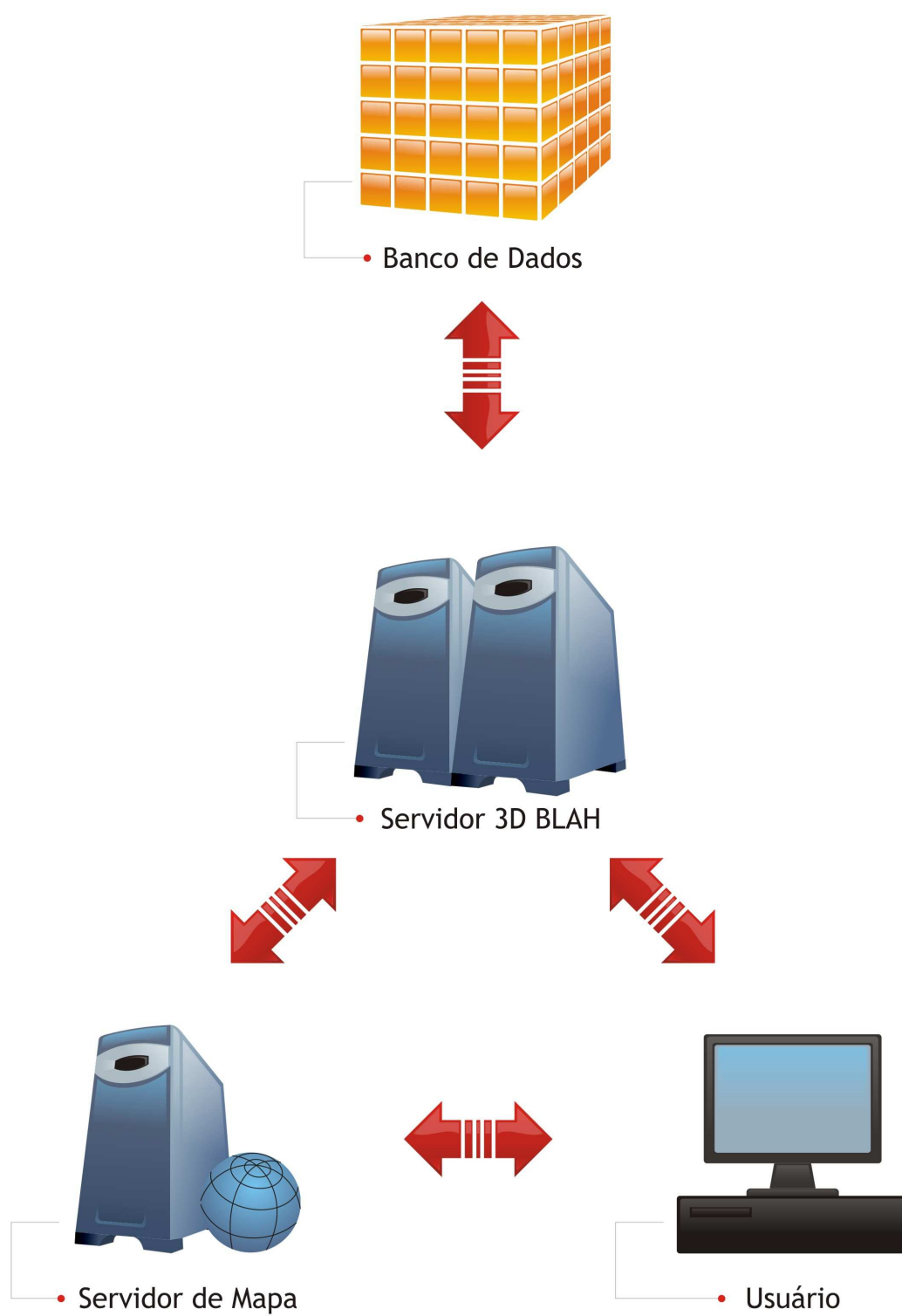
Esta tabela foi desenvolvida para facilitar a interpretação das mensagens enviadas e recebidas pelos servidores, facilitando assim a implementação do 3DBLAH. Ou seja, cada linha possui um protocolo que é uma mensagem vinda de um cliente a ser interpretada pelo servidor ou deferida pelo mesmo a um cliente.

Código	Tipo	Exemplo
001	Requisição de Login	001Usuario~Senha
002	Reprovação do Login	002MsgReprovação
003	Aprovação do Login	003ChaveUsuario~Nome~Nascimento~Sexo...
004	Adicionar Amigo pelo Conversa.exe	004ChaveUsuario
005	Alterar dados do usuário	005ChaveUsuario~Nome~Nascimento~Sexo...
006	Enviar Dados Atualizados Para Amigos	006ChaveUsuario~Nome~Nascimento~Sexo...
007	Requisição de Lista de Avatares	007
008	Lista de Avatares	008Nome1~Nome2r~Nome3~.....
009	Requisição de Lista de Amigos	009
010	Lista de Amigos	010Nome~Chave~Status~Sexo...
011	Adicionar Amigo pela Chave	011ChaveUsuario
012	Adicionar Amigo pelo E-mail	012Email
013	Remover Amigo	013ChaveAmigo
014	Amigo Adicionado	014ChaveUsuario~Apelido~Status~Sexo
015	Solicita Dados Contato ao Servidor	015ChaveContato
016	Solicitar Lista de Pedidos	016
017	Enviar Lista de Pedidos	017Chave~Email~Nome~50~bbb@ufpr.br~Maria...
018	Envia Dados Contato	018ChaveUsuario~Nome~Nascimento~Sexo...
019	Alterar Status do Usuário	0191
020	Enviar Novo Status Para Amigos	020001~1
021	Solicita Dados Contato	021ChaveContato
022	Enviar Novo Apelido Usuário P/ Conversa	022NovoNick
023	Enviar Novo Apelido Contato P/ Conversa	023ChaveContato~Nick
024	Envia Dados Contato	024ChaveUsuario~Nome~Nascimento~Sexo...
025	Enviar Msg	025ChaveUsuario~ChaveContato~Bom Dia!!!
026	Encaminhar Msg	026ChaveUsuario~ChaveContato~Bom Dia!!!
027	Encaminhar Msg	027ChaveUsuario~ChaveContato~Bom Dia!!!
028	Encaminhar Msg	028ChaveUsuario~ChaveContato~Bom Dia!!!
029	Forçar Foco em Conversa	029
030	Solicitar Página Servidores	0301
031	Enviar Dados Servidor	031Página~Posição~ChaveServidor~Título~Descrição...
032	Solicitar Vaga Em Servidor de Mapas	032ChaveUsuario~ChaveMod~Apelido~ChaveServidor~TamBsp

033	Conexão Com Servidor de Mapas Aceita	033
034	Conexão Com Servidor de Mapas Negada	034
035	Solicitar Dados Gerais	035
036	Enviar Dados Gerais	03615~QtdUsuarios
037	Solicitar Lista de Usuários no Mapa	037
038	Lista de Usuários	038ChaveUsuario~ArquivoModelo~Apelido~X~Y~Z~RY...
039	Avisa que Contato foi Adicionado à Lista	039
040	Requisição de Login	040Login~Senha
041	Aprovação do Login	041ChaveServidor~Titulo~Descrição~Capacidade...
042	Reprovação do Login	042MsgReprovação
043	Avisa que usuário x conectou	043ChaveUsuario~ArquivoModelo~Apelido~PosicaoX~...
044	Avisa que usuário x desconectou	044ChaveUsuario
045	Enviar Posição no Mapa	045x~y~z
046	Encaminha Posição Usuário no Mapa	046ChaveUsuario~x~y~z
047	Solicita Lista Arquivos	047
048	Kickar Usuário	048Motivo
048	Kickar Servidor de Mapas	048Motivo
049	Chamar a Atenção	049ChaveUsuario~ChaveContato
050	Encaminhar Nudge	050ChaveUsuario~ChaveContato
051	Encaminhar Nudge	050ChaveUsuario~ChaveContato
052	Encaminhar Nudge	050ChaveUsuario~ChaveContato
053	Ping Servidor Principal	053
054	Resposta do Ping	054
055	Solicita Envio de Arquivo	055ChaveUsuario~ChaveContato~NomeArquivo
056	Encaminhar Solicitação de Envio de Arquivo	056ChaveUsuario~ChaveContato~NomeArquivo
057	Avisa que Conversa.exe está pronto	057ChaveContato
058	Kickar Usuário	048Motivo
059	Solicitar Lista de Arquivos	059
060	Enviar Lista de Arquivos	060Nome~Email~NomeArquivo...
061	Enviar Nome Arquivo p/ Baixar	061NomeArquivo
062	Enviar Exclusão de Registro na Tabela	062NomeArquivo
063	Encaminhar Exclusao de Registro na Tabela	063NomeArquivo
064	Avisar Conversa que Contato Desconectou	064
065	Avisar Conversa que Contato Reconnectou	065

	Mensagens enviadas do 3DBLAH para o Servidor Principal
	Mensagens enviadas do Servidor Principal para o 3DBLAH
	Mensagens enviadas da janela de conversa para o 3DBLAH
	Mensagens enviadas do 3DBLAH para a janela de conversa
	Mensagens enviadas do 3DBLAH para o Servidor de Mapas
	Mensagens enviadas do Servidor de Mapas para o 3DBLAH

ANEXO III
ESQUEMA DO FUNCIONAMENTO BÁSICO DO 3DBLAH



ANEXO IV
SCRIPT DO BANCO DE DADOS

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[ArquivosEnviados]') and OBJECTPROPERTY(id,  
N'IsUserTable') = 1)  
drop table [dbo].[ArquivosEnviados]  
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Modelo]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)  
drop table [dbo].[Modelo]  
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Servidor]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)  
drop table [dbo].[Servidor]  
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Usuario]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)  
drop table [dbo].[Usuario]  
GO
```

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[UsuarioContato]') and OBJECTPROPERTY(id, N'IsUserTable') =  
1)  
drop table [dbo].[UsuarioContato]  
GO
```

```
CREATE TABLE [dbo].[ArquivosEnviados] (  
    [ChaveRemetente] [int] NOT NULL ,  
    [ChaveDestinatario] [int] NOT NULL ,
```

```

        [NomeArquivo] [varchar] (500) COLLATE Latin1_General_CI_AS NOT
NULL ,
        [Data] [datetime] NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[Modelo] (
        [ChaveModelo] [int] NOT NULL ,
        [NomeArquivo] [varchar] (50) COLLATE Latin1_General_CI_AS NULL ,
        [LinkDownload] [varchar] (500) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[Servidor] (
        [ChaveServidor] [int] NOT NULL ,
        [Senha] [varchar] (10) COLLATE Latin1_General_CI_AS NULL ,
        [Titulo] [varchar] (30) COLLATE Latin1_General_CI_AS NULL ,
        [Descricao] [varchar] (200) COLLATE Latin1_General_CI_AS NULL ,
        [Hostname] [varchar] (20) COLLATE Latin1_General_CI_AS NULL ,
        [Capacidade] [int] NULL ,
        [NomeArquivoMapa] [varchar] (50) COLLATE Latin1_General_CI_AS NULL
,
        [NomeArquivoTextura] [varchar] (50) COLLATE Latin1_General_CI_AS
NULL ,
        [LinkDownload] [varchar] (500) COLLATE Latin1_General_CI_AS NULL ,
        [Login] [varchar] (50) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[Usuario] (
        [ChaveUsuario] [int] NOT NULL ,

```



```

[Nome] [varchar] (50) COLLATE Latin1_General_CI_AS NULL ,
[Apelido] [varchar] (50) COLLATE Latin1_General_CI_AS NULL ,
[Nascimento] [datetime] NULL ,
[Sexo] [int] NULL ,
[Cidade] [varchar] (30) COLLATE Latin1_General_CI_AS NULL ,
[Estado] [varchar] (30) COLLATE Latin1_General_CI_AS NULL ,
[Pais] [int] NULL ,
[Email] [varchar] (50) COLLATE Latin1_General_CI_AS NULL ,
[Senha] [varchar] (8) COLLATE Latin1_General_CI_AS NULL ,
[EstadoCivil] [int] NULL ,
[ChaveModelo] [int] NULL ,
[StatusUsuario] [int] NULL ,
[Telefone] [varchar] (20) COLLATE Latin1_General_CI_AS NULL ,
[Celular] [varchar] (20) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[UsuarioContato] (
    [ChaveUsuario] [int] NOT NULL ,
    [ChaveAmigo] [int] NOT NULL ,
    [Pendente] [bit] NULL
) ON [PRIMARY]
GO

```

```

ALTER TABLE [dbo].[ArquivosEnviados] WITH NOCHECK ADD
    CONSTRAINT [PK_ArquivosEnviados] PRIMARY KEY CLUSTERED
    (
        [ChaveRemetente],
        [ChaveDestinatario],
        [NomeArquivo]
    ) ON [PRIMARY]

```

GO

```
ALTER TABLE [dbo].[Modelo] WITH NOCHECK ADD
    CONSTRAINT [PK_Modelo] PRIMARY KEY CLUSTERED
    (
        [ChaveModelo]
    ) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[Servidor] WITH NOCHECK ADD
    CONSTRAINT [PK_Servidor] PRIMARY KEY CLUSTERED
    (
        [ChaveServidor]
    ) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[Usuario] WITH NOCHECK ADD
    CONSTRAINT [DF_Usuario_Pais] DEFAULT (26) FOR [Pais],
    CONSTRAINT [DF_Usuario_ChaveModelo] DEFAULT (1) FOR
    [ChaveModelo],
    CONSTRAINT [DF_Usuario_StatusUsuario] DEFAULT ((-1)) FOR
    [StatusUsuario],
    CONSTRAINT [PK_Usuario] PRIMARY KEY CLUSTERED
    (
        [ChaveUsuario]
    ) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[UsuarioContato] WITH NOCHECK ADD
    CONSTRAINT [PK_UsuarioContato] PRIMARY KEY CLUSTERED
    (
```

```
        [ChaveUsuario],  
        [ChaveAmigo]  
    ) ON [PRIMARY]  
GO
```

ANEXO V
CODIFICAÇÃO DO 3D BLAH

```
unit uAdicionarContato;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, uConfiguracoes, uFuncoes, uConstantes;
```

```
type
```

```
TfAdicionarContato = class(TForm)
```

```
    btAdicionar: TBitBtn;
```

```
    btCancelar: TBitBtn;
```

```
    lbEmail: TLabel;
```

```
    edEmail: TEdit;
```

```
    procedure btAdicionarClick(Sender: TObject);
```

```
    procedure btCancelarClick(Sender: TObject);
```

```
private
```

```
    { Private declarations }
```

```
public
```

```
    function AbrirForm(var email: string): boolean;
```

```
    { Public declarations }
```

```
end;
```

```
var
```

```
    fAdicionarContato: TfAdicionarContato;
```

```
implementation
```

```
{ $R *.dfm }
```

```
{ TfAdicionarContato }
```

```

function TfAdicionarContato.AbrirForm(var email: string): boolean;
begin
    showmodal;
    if (modalresult=mrok) then
        result := true
    else
        result := false;
    email := RemoveCaracteres(edEmail.Text, [Separador, SeparadorTamanho,
SubstitutoEspaco]);
end;

procedure TfAdicionarContato.btAdicionarClick(Sender: TObject);
begin
    modalresult := mrok;
end;

procedure TfAdicionarContato.btCancelarClick(Sender: TObject);
begin
    modalresult := mrcancel;
end;

end.

unit uBaixaArquivo;

interface

uses
    Classes, IdComponent, IdTCPConnection, IdTCPClient, IdFTP, uFTP, SysUtils,

```

uFuncoes, ExtCtrls, Graphics, Dialogs, ComCtrls;

type

 TBaixaArquivo = class(TThread)

 private

 Diretorios: TStringList;

 Origem: string;

 Arquivo: string;

 Progresso: TProgressBar;

 { Private declarations }

 public

 IdFTP: TIdFTP;

 constructor Criar(ArquivoOrigem: string; NomeArquivo: string; Barra:
TProgressBar);

 destructor Destroy; override;

 procedure IdFTPWork(ASender: TObject; AWorkMode: TWorkMode;
AWorkCount: Integer);

 protected

 procedure Execute; override;

 end;

implementation

{ TBaixaArquivo }

 constructor TBaixaArquivo.Criar(ArquivoOrigem: string; NomeArquivo: string;
Barra: TProgressBar);

 begin

 Create(true);

 IdFTP := TIdFTP.Create(nil);

 Diretorios := TStringList.Create;

```

FreeOnTerminate := true;
Origem := ArquivoOrigem;
Arquivo := NomeArquivo;
Progresso := Barra;
IdFTP.OnWork := IdFTPWork;
end;

destructor TBaixaArquivo.Destroy;
begin
    freeandnil(IdFTP);
    freeandnil(Diretorios);
    inherited;
end;

procedure TBaixaArquivo.Execute;
var x: integer;
begin
    try
        IdFTP.Username := FTP.FTPUser;
        IdFTP.Password := FTP.FTPPass;
        IdFTP.Host := FTP.FTPHost;
        IdFTP.Connect;
        GetDirList(FTP.FTPDir, Diretorios);
        for x := 0 to Diretorios.Count-1 do
            IdFTP.ChangeDir(Diretorios[x]);
            IdFTP.ChangeDir(FTP.FTPFileDir);
            Progresso.Max := IdFTP.Size(Origem);
            if (FileExists(Arquivo)) then
                DeleteFile(Arquivo);
            IdFTP.Get(Origem, Arquivo, false);
        except

```



```

        showmessage('Erro na transferência do arquivo!');
    end;
    IdFTP.Disconnect;
end;

```

```

procedure TBaixaArquivo.IdFTPWork(ASender: TObject; AWorkMode:
TWorkMode; AWorkCount: Integer);
begin
    Progresso.Position := AWorkCount;
end;

end.

```

```

unit uBaixaImgMapa;

```

```

interface

```

```

uses

```

```

    Classes, IdComponent, IdTCPConnection, IdTCPClient, IdFTP, uFTP, SysUtils,
    uConfiguracoes, uFuncoes, ExtCtrls;

```

```

type

```

```

    TBaixaImgMapa = class(TThread)

```

```

    private

```

```

        IdFTP: TIdFTP;

```

```

        Diretorios: TStringList;

```

```

        Img: TImage;

```

```

        { Private declarations }

```

```

    public

```

```

        ChaveServidor: integer;

```

```

    constructor Criar(Image: TImage);
    destructor Destroy; override;
protected
    procedure Execute; override;
end;

```

implementation

{ Important: Methods and properties of objects in visual components can only be used in a method called using Synchronize, for example,

```

    Synchronize(UpdateCaption);

```

and UpdateCaption could look like,

```

    procedure TBaixaImgMapa.UpdateCaption;
    begin
        Form1.Caption := 'Updated in a thread';
    end; }

```

{ TBaixaImgMapa }

```

constructor TBaixaImgMapa.Criar(Image: TImage);
begin
    Create(true);
    IdFTP := TIdFTP.Create(nil);
    Diretorios := TStringList.Create;
    FreeOnTerminate := true;
    Img := Image;
end;

```

```

destructor TBaixalmgMapa.Destroy;
begin
  freeandnil(IdFTP);
  freeandnil(Diretorios);
  inherited;
end;

procedure TBaixalmgMapa.Execute;
var Arquivo: string;
    x: integer;
begin
  Arquivo :=
configuracoes.Dir_Thumbs+FormatFloat('000000',ChaveServidor)+'.jpg';
  try
    IdFTP.Username := FTP.FTPUser;
    IdFTP.Password := FTP.FTPPass;
    IdFTP.Host := FTP.FTPHost;
    IdFTP.Connect;
    GetDirList(FTP.FTPDir, Diretorios);
    for x := 0 to Diretorios.Count-1 do
      IdFTP.ChangeDir(Diretorios[x]);
      IdFTP.ChangeDir(FTP.FTPThumbDir);
      if (not FileExists(Arquivo)) or
      ((FileSize(Arquivo)<>IdFTP.Size(FormatFloat('000000',ChaveServidor)+'.jpg')) and
      (IdFTP.Size(FormatFloat('000000',ChaveServidor)+'.jpg')>0)) then
        begin
          if (FileExists(Arquivo)) then
            DeleteFile(Arquivo);
            IdFTP.Get(FormatFloat('000000',ChaveServidor)+'.jpg', Arquivo, false);
        end;
      if FileExists(Arquivo) and (not Terminated) then

```

```

        Img.Picture.LoadFromFile(Arquivo);
    finally
        IdFTP.Disconnect;
    end;
end;

end.

```

```

unit uBaixaImgUsuario;

```

```

interface

```

```

uses

```

```

    Classes, IdComponent, IdTCPConnection, IdTCPClient, IdFTP, uFTP, SysUtils,
    uConfiguracoes, uFuncoes, ExtCtrls, Graphics;

```

```

type

```

```

    TBaixaImgUsuario = class(TThread)

```

```

    private

```

```

        IdFTP: TIdFTP;

```

```

        Diretorios: TStringList;

```

```

        Picture: TPicture;

```

```

        Img: TImage;

```

```

        { Private declarations }

```

```

    public

```

```

        ChaveUsuario: integer;

```

```

        constructor Criar(Pic: TPicture; Image: TImage);

```

```

        destructor Destroy; override;

```

```

    protected

```

```

        procedure Execute; override;

```

end;

implementation

{ TBaixaImgUsuario }

constructor TBaixaImgUsuario.Criar(Pic: TPicture; Image: TImage);

begin

 Create(true);

 IdFTP := TIdFTP.Create(nil);

 Diretorios := TStringList.Create;

 FreeOnTerminate := true;

 Picture := Pic;

 Img := Image;

end;

destructor TBaixaImgUsuario.Destroy;

begin

 freeandnil(IdFTP);

 freeandnil(Diretorios);

 inherited;

end;

procedure TBaixaImgUsuario.Execute;

var Arquivo: string;

 x: integer;

begin

 Arquivo := configuracoes.Dir_Images+FormatFloat('000000',ChaveUsuario)+''.jpg';

 try

 IdFTP.Username := FTP.FTPUser;

 IdFTP.Password := FTP.FTPPass;

```

IdFTP.Host := FTP.FTPHost;
IdFTP.Connect;
GetDirList(FTP.FTPDir, Diretorios);
for x := 0 to Diretorios.Count-1 do
  IdFTP.ChangeDir(Diretorios[x]);
IdFTP.ChangeDir(FTP.FTPImgDir);
if (not FileExists(Arquivo)) or
((FileSize(Arquivo)<>IdFTP.Size(FormatFloat('000000',ChaveUsuario)+'.jpg')) and
(IdFTP.Size(FormatFloat('000000',ChaveUsuario)+'.jpg')>0)) then
  begin
    if (FileExists(Arquivo)) then
      DeleteFile(Arquivo);
    IdFTP.Get(FormatFloat('000000',ChaveUsuario)+'.jpg', Arquivo, false);
  end;
if FileExists(Arquivo) and (not Terminated) then
  begin
    Picture.LoadFromFile(Arquivo);
    if (Img<>nil) then
      Img.Picture := Picture;
  end;
finally
  IdFTP.Disconnect;
end;
end;

end.

```

```

unit uConfiguracoes;
{
  le e armazena configuracoes do sistema.

```

para adicionar outro parametro de configuração, basta adicionar uma propriedade a classe, na sessao published, que ela vai ser automaticamente armazenada.

Para adicionar uma propriedade, basta adicionar a linha:

```
property NomeDaPropriedade:Tipo;
```

e usar a função 'Complete class at cursor' (Ctrl+Shift+C), para criar automaticamente a função setPropriedade e a propriedade private correspondente automaticamente.

```
}
```

```
interface
```

```
uses
```

```
graphics, controls, classes, activex, forms;
```

```
type
```

```
TConfiguracoes=class(TComponent)
```

```
private
```

```
Arquivo: string;
```

```
FServerHost: string;
```

```
FSomLogin: boolean;
```

```
FSomMsg: boolean;
```

```
FSomNudge: boolean;
```

```
FSomConsole: boolean;
```

```
FSomScreenshot: boolean;
```

```
FSkin: boolean;
```

```
FSkinDialogo: boolean;
```

```

FTecla_Up: TOleEnum;
FTecla_Left: TOleEnum;
FTecla_Right: TOleEnum;
FTecla_Down: TOleEnum;
FDir_Images: string;
FDir_Thumbs: string;
FDir_Screenshots: string;
FLingua: integer;
FLights: boolean;
procedure SetServerHost(const Value: string);
procedure SetSomLogin(const Value: boolean);
procedure SetSomMsg(const Value: boolean);
procedure SetSomNudge(const Value: boolean);
procedure SetSomConsole(const Value: boolean);
procedure SetSomScreenshot(const Value: boolean);
procedure SetSkin(const Value: boolean);
procedure SetSkinDialogo(const Value: boolean);
procedure SetTecla_Up(const Value: TOleEnum);
procedure SetTecla_Down(const Value: TOleEnum);
procedure SetTecla_Left(const Value: TOleEnum);
procedure SetTecla_Right(const Value: TOleEnum);
procedure SetDir_Images(const Value: string);
procedure SetDir_Screenshots(const Value: string);
procedure SetDir_Thumbs(const Value: string);
procedure SetLingua(const Value: integer);
procedure SetLights(const Value: boolean);
published
property Lingua: integer read FLingua write SetLingua;
property ServerHost: string read FServerHost write SetServerHost;
property Skin: boolean read FSkin write SetSkin;
property SkinDialogo: boolean read FSkinDialogo write SetSkinDialogo;

```



```

property SomLogin: boolean read FSomLogin write SetSomLogin;
property SomMsg: boolean read FSomMsg write SetSomMsg;
property SomNudge: boolean read FSomNudge write SetSomNudge;
property SomConsole: boolean read FSomConsole write SetSomConsole;
property SomScreenshot: boolean read FSomScreenshot write
SetSomScreenshot;
property Tecla_Up: TOleEnum read FTecla_Up write SetTecla_Up;
property Tecla_Down: TOleEnum read FTecla_Down write SetTecla_Down;
property Tecla_Right: TOleEnum read FTecla_Right write SetTecla_Right;
property Tecla_Left: TOleEnum read FTecla_Left write SetTecla_Left;
property Dir_Images: string read FDir_Images write SetDir_Images;
property Dir_Thumbs: string read FDir_Thumbs write SetDir_Thumbs;
property Dir_Screenshots: string read FDir_Screenshots write
SetDir_Screenshots;
property Lights: boolean read FLights write SetLights;
public
constructor Create(aOwner:tcomponent);override;
destructor Destroy;override;

procedure Carregar;
procedure Gravar;
procedure RestaurarPadrao;
end;

var
    Configuracoes:tconfiguracoes;

implementation

uses sysutils, uMensagemStatus, uFuncoes;

```

```
{ TConfiguracoes }
```

```
procedure TConfiguracoes.Carregar;
```

```
begin
```

```
    NovaMsgStatus('Carregando configurações...');
```

```
    try
```

```
        LerObjetoComoTexto(arquivo,self);
```

```
    finally
```

```
        RemoverMsgStatus;
```

```
    end;
```

```
end;
```

```
constructor TConfiguracoes.Create(aOwner:tcomponent);
```

```
begin
```

```
    inherited;
```

```
    restaurarPadrao;
```

```
end;
```

```
destructor TConfiguracoes.Destroy;
```

```
begin
```

```
    inherited;
```

```
end;
```

```
procedure TConfiguracoes.Gravar;
```

```
begin
```

```
    GravarObjetoComoTexto(arquivo,self);
```

```
end;
```

```
procedure TConfiguracoes.RestaurarPadrao;
```

```
begin
```

```
    //valores padrão
```

```

Lingua := 1;
ServerHost := 'Time';
Skin := true;
SkinDialogo := true;
SomLogin := true;
SomMsg := true;
SomNudge := true;
SomConsole := true;
SomScreenshot := true;
Tecla_UP := $00000011;
Tecla_Down := $0000001F;
Tecla_Right := $00000020;
Tecla_Left := $0000001E;
Dir_Images := extractfilepath(application.ExeName)+'Images\';
Dir_Thumbs := extractfilepath(application.ExeName)+'Thumbs\';
Dir_Screenshots := extractfilepath(application.ExeName)+'Screenshots\';
Lights := true;
end;

```

```

procedure TConfiguracoes.SetDir_Images(const Value: string);
begin
    FDir_Images := Value;
end;

```

```

procedure TConfiguracoes.SetDir_Screenshots(const Value: string);
begin
    FDir_Screenshots := Value;
end;

```

```

procedure TConfiguracoes.SetDir_Thumbs(const Value: string);
begin

```

```
    FDir_Thumbs := Value;  
end;
```

```
procedure TConfiguracoes.SetLights(const Value: boolean);  
begin  
    FLights := Value;  
end;
```

```
procedure TConfiguracoes.SetLingua(const Value: integer);  
begin  
    FLingua := Value;  
end;
```

```
procedure TConfiguracoes.SetServerHost(const Value: string);  
begin  
    FServerHost := Value;  
end;
```

```
procedure TConfiguracoes.SetSkin(const Value: boolean);  
begin  
    FSkin := Value;  
end;
```

```
procedure TConfiguracoes.SetSkinDialogo(const Value: boolean);  
begin  
    FSkinDialogo := Value;  
end;
```

```
procedure TConfiguracoes.SetSomConsole(const Value: boolean);  
begin  
    FSomConsole := Value;
```

end;

```
procedure TConfiguracoes.SetSomLogin(const Value: boolean);  
begin  
    FSomLogin := Value;  
end;
```

```
procedure TConfiguracoes.SetSomMsg(const Value: boolean);  
begin  
    FSomMsg := Value;  
end;
```

```
procedure TConfiguracoes.SetSomNudge(const Value: boolean);  
begin  
    FSomNudge := Value;  
end;
```

```
procedure TConfiguracoes.SetSomScreenshot(const Value: boolean);  
begin  
    FSomScreenshot := Value;  
end;
```

```
procedure TConfiguracoes.SetTecla_Down(const Value: TOleEnum);  
begin  
    FTecla_Down := Value;  
end;
```

```
procedure TConfiguracoes.SetTecla_Left(const Value: TOleEnum);  
begin  
    FTecla_Left := Value;  
end;
```

```

procedure TConfiguracoes.SetTecla_Right(const Value: TOleEnum);
begin
    FTecla_Right := Value;
end;

```

```

procedure TConfiguracoes.SetTecla_Up(const Value: TOleEnum);
begin
    FTecla_Up := Value;
end;

```

```

initialization
    configuracoes:=tconfiguracoes.create(nil);
    try
        configuracoes.arquivo:='Config.cfg';
        configuracoes.carregar;
    except
    end;
finalization
    configuracoes.free;
end.

```

```

unit uConstantes;

```

```

interface

```

```

const
    Separador = 'µ';
    SeparadorTamanho = '∅';
    SubstitutoEspaco = '±';

```

implementation

end.

unit uDtModule;

interface

uses

SysUtils, Classes, ZConnection, ScktComp, uFuncoes, DB,
uTolderlundStringGrid,
ZAbstractRODataset, ZAbstractDataset, ZDataset, uConstantes, uMapa;

type

TDtModule = class(TDataModule)

Connection: TZConnection;

Server: TServerSocket;

procedure ServerClientConnect(Sender: TObject; Socket: TCustomWinSocket);

procedure ServerClientDisconnect(Sender: TObject; Socket:

TCustomWinSocket);

procedure ServerClientRead(Sender: TObject; Socket: TCustomWinSocket);

procedure DataModuleCreate(Sender: TObject);

procedure DataModuleDestroy(Sender: TObject);

procedure ServerClientError(Sender: TObject; Socket: TCustomWinSocket;

ErrorEvent: TErrorEvent; var ErrorCode: Integer);

private

{ Private declarations }

public

Avatares: string;

```

Servidores: array of TMapa;
IDConexoes: TList; //Gerenciador de conexões
ChavesUsuarios: TStringList;
TiposUsuarios: TStringList;
procedure InterpretaMsg(Codigo: integer; Msg: string; Cliente: Pointer);
procedure EfetuaLoginUsuario(Msg: string; Cliente: Pointer);
procedure EnviaListaAvatares(Msg: string; Cliente: Pointer);
procedure EnviaListaContatos(Msg: string; Cliente: Pointer);
procedure AdicionaContatoPelaChave(Msg: string; Cliente: Pointer);
procedure AdicionaContatoPeloEmail(Msg: string; Cliente: Pointer);
procedure RemoveContatoDeUsuario(Msg: string; Cliente: Pointer);
procedure AtualizaDadosUsuario(Msg: string; Cliente: Pointer);
procedure AtualizaStatusUsuario(Msg: string; Cliente: Pointer);
procedure EnviaListaPedidosAmigos(Msg: string; Cliente: Pointer);
procedure EnviaListaArquivos(Msg: string; Cliente: Pointer);
procedure ExclueRegistroArquivoEnviado(Msg: string; Cliente: Pointer);
procedure EncaminhaMsg(Msg: string; Cliente: Pointer);
procedure EncaminhaNudge(Msg: string; Cliente: Pointer);
procedure EnviaDadosGerais(Msg: string; Cliente: Pointer);
procedure EnviaPaginaServidores(Msg: string; Cliente: Pointer);
procedure EfetuaLoginMapa(Msg: string; Cliente: Pointer);
procedure RespondePing(Msg: string; Cliente: Pointer);
procedure EnviaDadosContato(Msg: string; Cliente: Pointer);
procedure RegistraEnvioDeArquivo(Msg: string; Cliente: Pointer);
procedure PreencheListaAvatares();
procedure PreencheListaServidores();
function LocalizaCliente(Cliente: Pointer): integer;
procedure SetarStatusUsuario(ChaveUsuario: integer; Status: integer);
procedure ListarAmigosUsuario(ChaveUsuario: integer; var Qry: TZQuery);
procedure KickarUsuario(Index: integer; Motivo: string);
procedure SetarTodosUsuariosOffline();

```



```

    { Public declarations }
end;

var
    DtModule: TDtModule;

implementation

uses ufsPrincipal;

{$R *.dfm}

// -----
//  Eventos dos componentes
// -----

procedure TDtModule.ServerClientConnect(Sender: TObject; Socket:
TCustomWinSocket);
var IDsocket: ^byte;
begin
    New(IDSocket);
    Socket.Data := IDSocket;
    IDConexoes.Add(Socket.data);
    ChavesUsuarios.Add('0');
    TiposUsuarios.Add('');
    fsPrincipal.Grid.RowCount := Server.Socket.ActiveConnections+1;
    fsPrincipal.Grid.Cells[0,Server.Socket.ActiveConnections] := Socket.RemoteHost;
    fsPrincipal.Grid.Cells[2,Server.Socket.ActiveConnections] := datetostr(now)+' -
'+timetostr(now);
end;

```

```

procedure TDtModule.ServerClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
var Index: integer;
    Chave: integer;
begin
    Index := LocalizaCliente(Socket.Data);
    Chave := strtoint(ChavesUsuarios[Index]);
    if (TiposUsuarios.Strings[Index]='Usuario') then
        SetarStatusUsuario(Chave, -1);
    ChavesUsuarios.Delete(Index);
    TiposUsuarios.Delete(Index);
    TStringGridHack(fsPrincipal.Grid).DeleteRow(Index+1);
    IDConexoes.Remove(Socket.Data);
    Dispose(Socket.Data);
end;

```

```

procedure TDtModule.ServerClientRead(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Cliente: Pointer;
    Lista: TStringList;
    x: integer;
begin
    Cliente := Socket.Data;
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x], Codigo, Msg);
            end;
        end;
    except
        ;
    end;
end;

```

```
    InterpretaMsg(Codigo, Msg, Cliente);  
end;  
finally  
    freeandnil(Lista);  
end;  
end;
```

```
procedure TDtModule.ServerClientError(Sender: TObject; Socket:  
TCustomWinSocket; ErrorEvent: TErrorEvent;  
    var ErrorCode: Integer);  
begin  
    ErrorCode := 0;  
end;
```

```
procedure TDtModule.DataModuleCreate(Sender: TObject);  
begin  
    SetarTodosUsuariosOffline();  
    IDConexoes := TList.Create;  
    IDConexoes.Clear;  
    ChavesUsuarios := TStringList.Create();  
    TiposUsuarios := TStringList.Create();  
    ChavesUsuarios.Clear;  
    TiposUsuarios.Clear;  
end;
```

```
procedure TDtModule.DataModuleDestroy(Sender: TObject);  
begin  
    freeandnil(IDConexoes);  
    freeandnil(ChavesUsuarios);  
    freeandnil(TiposUsuarios);  
end;
```

```

// -----
//  Eventos dos componentes
// -----

// -----
//  Trata as mensagens recebidas dos clientes
// -----

procedure TDtModule.InterpretaMsg(Codigo: integer; Msg: string; Cliente: Pointer);
begin
  if (Codigo=01) then
    EfetuaLoginUsuario(Msg, Cliente)
  else if (Codigo=05) then
    AtualizaDadosUsuario(Msg, Cliente)
  else if (Codigo=07) then
    EnviaListaAvatares(Msg, Cliente)
  else if (Codigo=09) then
    EnviaListaContatos(Msg, Cliente)
  else if (Codigo=11) then
    AdicionaContatoPelaChave(Msg, Cliente)
  else if (Codigo=12) then
    AdicionaContatoPeloEmail(Msg, Cliente)
  else if (Codigo=13) then
    RemoveContatoDeUsuario(Msg, Cliente)
  else if (Codigo=15) then
    EnviaDadosContato(Msg, Cliente)
  else if (Codigo=16) then
    EnviaListaPedidosAmigos(Msg, Cliente)
  else if (Codigo=19) then
    AtualizaStatusUsuario(Msg, Cliente)

```

```

else if (Codigo=26) then
    EncaminhaMsg(Msg, Cliente)
else if (Codigo=30) then
    EnviaPaginaServidores(Msg, Cliente)
else if (Codigo=35) then
    EnviaDadosGerais(Msg, Cliente)
else if (Codigo=40) then
    EfetuaLoginMapa(Msg, Cliente)
else if (Codigo=50) then
    EncaminhaNudge(Msg, Cliente)
else if (Codigo=53) then
    RespondePing(Msg, Cliente)
else if (Codigo=56) then
    RegistraEnvioDeArquivo(Msg, Cliente)
else if (Codigo=59) then
    EnviaListaArquivos(Msg, Cliente)
else if (Codigo=63) then
    ExclueRegistroArquivoEnviado(Msg, Cliente);
end;

procedure TDtModule.EfetuaLoginUsuario(Msg: string; Cliente: Pointer);
var Lista: TStringList;
    Qry: TZQuery;
    chave_usuario: integer;
    email: string;
    nome: string;
    apelido: string;
    nascimento: tdatetime;
    sexo: integer;
    estado_civil: integer;
    cidade: string;

```

```

    estado: string;
    pais: string;
    telefone: string;
    celular: string;
    chave_modelo: integer;
    nome_arquivo: string;
    index: integer;
    Chave: integer;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    Qry := TZQuery.Create(nil);
    Qry.Connection := Connection;
    Qry.SQL.Add('Select * from Usuario U left join Modelo M on
U.ChaveModelo=M.ChaveModelo');
    Qry.SQL.Add('where email=:usuario');
    Qry.ParamByName('usuario').AsString := Lista.Strings[0];
    Qry.Open;
    if (Qry.RecordCount=0) then
        Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('002E-
mail não cadastrado!'))
    else
        begin
            nome := Lista.Strings[1];
            if (Qry.FieldName('Senha').AsString=Lista.Strings[1]) then
                begin
                    if (Qry.fieldbyname('StatusUsuario').AsInteger<>-1) then

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('002Este
usuário já está conectado!'))
                else

```

```

begin
    chave_usuario := Qry.fieldbyname('ChaveUsuario').AsInteger;
    email := Qry.fieldbyname('Email').AsString;
    nome := Qry.fieldbyname('Nome').AsString;
    apelido := Qry.fieldbyname('Apelido').AsString;
    nascimento := Qry.fieldbyname('Nascimento').AsDateTime;
    sexo := Qry.fieldbyname('Sexo').AsInteger;
    estado_civil := Qry.fieldbyname('EstadoCivil').AsInteger;
    cidade := Qry.fieldbyname('Cidade').AsString;
    estado := Qry.fieldbyname('Estado').AsString;
    pais := Qry.fieldbyname('Pais').AsString;
    telefone := Qry.fieldbyname('Telefone').AsString;
    celular := Qry.fieldbyname('Celular').AsString;
    chave_modelo := Qry.fieldbyname('ChaveModelo').AsInteger;
    nome_arquivo := Qry.fieldbyname('NomeArquivo').AsString;
    Index := IDConexoes.IndexOf(Cliente);
    ChavesUsuarios.Strings[Index] := inttostr(chave_usuario);
    TiposUsuarios.Strings[Index] := 'Usuario';
    fsPrincipal.Grid.Cells[3,Index+1] := 'Usuario';
    fsPrincipal.Grid.Cells[1,Index+1] := Lista.Strings[0];

```

```

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('003'+intt
ostr(chave_usuario)+separador+

```

```

    nome+separador+
    apelido+separador+
    datetostr(nascimento)+separador+
    inttostr(sexo)+separador+
    inttostr(estado_civil)+separador+
    cidade+separador+
    estado+separador+
    pais+separador+

```

```

        telefone+separador+
        celular+separador+
        '4'+separador+
        inttostr(chave_modelo)+separador+
        nome_arquivo+separador+
        email));

    Qry.SQL.Clear;
    Qry.SQL.Add('Update Usuario set StatusUsuario=4 where
ChaveUsuario=:Chave');
    Qry.ParamByName('Chave').AsInteger := chave_usuario;
    Qry.ExecSQL;
    ListarAmigosUsuario(chave_usuario, Qry);
    Qry.First;
    while not Qry.Eof do
    begin
        Chave := Qry.fieldbyname('Chave').AsInteger;
        Index := ChavesUsuarios.IndexOf(inttostr(Chave));
        if (Index<>-1) then

DtModule.Server.Socket.Connections[Index].SendText(Protocolo('020'+inttostr(chave_usuario)+separador+'4'));

        Qry.Next;
    end;
end;
end
else

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('002Senha incorreta!'));

end;
freeandnil(Qry);

```



```

    freeandnil(Lista);
end;

procedure TDtModule.AtualizaDadosUsuario(Msg: string; Cliente: Pointer);
var Lista: TStringList;
    Qry: TZQuery;
    Chave: integer;
    Index: integer;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Update Usuario set ');
    Qry.SQL.Add('nome=:nome,');
    Qry.SQL.Add('apelido=:apelido,');
    Qry.SQL.Add('nascimento=:nascimento,');
    Qry.SQL.Add('sexo=:sexo,');
    Qry.SQL.Add('estadocivil=:estadocivil,');
    Qry.SQL.Add('cidade=:cidade,');
    Qry.SQL.Add('estado=:estado,');
    Qry.SQL.Add('pais=:pais,');
    Qry.SQL.Add('telefone=:telefone,');
    Qry.SQL.Add('celular=:celular,');
    Qry.SQL.Add('statususuario=:status,');
    Qry.SQL.Add('chavemodelo=:chavemodelo');
    Qry.SQL.Add('where chaveusuario=:chaveusuario');
    Qry.ParamByName('nome').AsString := Lista.Strings[1];
    Qry.ParamByName('apelido').AsString := Lista.Strings[2];
    Qry.ParamByName('nascimento').AsDate := strtodate(Lista.Strings[3]);
    Qry.ParamByName('sexo').AsString := Lista.Strings[4];

```

```

Qry.ParamByName('estadocivil').AsString := Lista.Strings[5];
Qry.ParamByName('cidade').AsString := Lista.Strings[6];
Qry.ParamByName('estado').AsString := Lista.Strings[7];
Qry.ParamByName('pais').AsString := Lista.Strings[8];
Qry.ParamByName('telefone').AsString := Lista.Strings[9];
Qry.ParamByName('celular').AsString := Lista.Strings[10];
Qry.ParamByName('status').AsString := Lista.Strings[11];
Qry.ParamByName('chavemodelo').AsString := Lista.Strings[12];
Qry.ParamByName('chaveusuario').AsString := Lista.Strings[0];
Qry.ExecSQL;

```

```

ListarAmigosUsuario(strtoint(Lista.Strings[0]), Qry);
Qry.First;
while not Qry.Eof do
begin
    Chave := Qry.fieldbyname('Chave').AsInteger;
    Index := ChavesUsuarios.IndexOf(inttostr(Chave));
    if (Index<>-1) then
        DtModule.Server.Socket.Connections[Index].SendText(Protocolo('006'+Msg));
    Qry.Next;
end;

```

```

freeandnil(Qry);
freeandnil(Lista);
end;

```

```

procedure TDtModule.EnviaListaAvatares(Msg: string; Cliente: Pointer);
begin

```

```

    Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('008'+Avatares))

```

end;

```

procedure TDtModule.EnviaListaContatos(Msg: string; Cliente: Pointer);
var Contatos: string;
    Qry: TZQuery;
    Index: integer;
begin
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select U.ChaveUsuario, U.Apelido, U.StatusUsuario, U.Sexo,
U.Email, U.Pais from Usuario U');
    Qry.SQL.Add('where U.ChaveUsuario in');
    Qry.SQL.Add('((Select UC.ChaveUsuario from UsuarioContato UC');
    Qry.SQL.Add('where (UC.ChaveAmigo=:Chave) and
(UC.Pendente=:Pendente)))');
    Qry.SQL.Add('Union');
    Qry.SQL.Add('(Select UC.ChaveAmigo from UsuarioContato UC');
    Qry.SQL.Add('where (UC.ChaveUsuario=:Chave) and
(UC.Pendente=:Pendente)))');
    Qry.SQL.Add('order by U.Apelido');
    Index := IDConexoes.IndexOf(Cliente);
    Qry.ParamByName('Chave').AsString := ChavesUsuarios.Strings[Index];
    Qry.ParamByName('Pendente').AsBoolean := false;
    Qry.Open;
    Qry.First;
    while (not Qry.Eof) do
    begin
        Contatos := Contatos+Qry.fieldbyname('Apellido').AsString+Separador+
            Qry.fieldbyname('ChaveUsuario').AsString+Separador+
            Qry.fieldbyname('StatusUsuario').AsString+Separador+
            Qry.fieldbyname('Sexo').AsString+Separador+

```

```

        Qry.fieldbyname('Email').AsString+Separador+
        Qry.fieldbyname('Pais').AsString+Separador;
    Qry.Next();
end;

```

```

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('010'+Co
ntatos));
    freeandnil(Qry);
end;

```

```

procedure TDtModule.RemoveContatoDeUsuario(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Index: integer;
begin
    Qry := TZQuery.Create(nil);
    Qry.Connection := dtModule.Connection;
    Qry.SQL.Add('Delete from UsuarioContato');
    Qry.SQL.Add('where (ChaveUsuario=:ChaveUsuario and
ChaveAmigo=:ChaveContato)');
    Qry.SQL.Add('or (ChaveUsuario=:ChaveContato and
ChaveAmigo=:ChaveUsuario)');
    Index := IDConexoes.IndexOf(Cliente);
    Qry.ParamByName('ChaveUsuario').AsString := ChavesUsuarios[Index];
    Qry.ParamByName('ChaveContato').AsString := Msg;
    Qry.ExecSQL;
    freeandnil(Qry);
end;

```

```

procedure TDtModule.AdicionaContatoPeloEmail(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
begin

```

```

Qry := TZQuery.Create(nil);
Qry.Connection := dtModule.Connection;
Qry.SQL.Add('Select ChaveUsuario from Usuario');
Qry.SQL.Add('where Email=:Email');
Qry.ParamByName('Email').AsString := Msg;
Qry.Open;
if (Qry.RecordCount>0) then
    AdicionaContatoPelaChave(Qry.fieldbyname('ChaveUsuario').AsString, Cliente);
    freeandnil(Qry);
end;

```

```

procedure TDtModule.AtualizaStatusUsuario(Msg: string; Cliente: Pointer);
var ChaveUsuario: integer;
begin
    ChaveUsuario := strtoint(ChavesUsuarios[LocalizaCliente(Cliente)]);
    SetarStatusUsuario(ChaveUsuario, strtoint(Msg));
end;

```

```

procedure TDtModule.EnviaListaPedidosAmigos(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Index: integer;
    Chave: integer;
    Lista: string;
begin
    Lista := "";
    Index := IDConexoes.IndexOf(Cliente);
    Chave := strtoint(ChavesUsuarios[Index]);
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select U.ChaveUsuario, U.Nome, U.Email');
    Qry.SQL.Add('from Usuario U where U.ChaveUsuario in');

```

```

    Qry.SQL.Add('(Select ChaveUsuario from UsuarioContato where
(ChaveAmigo=:Chave) and (Pendente=:Pendente))');
    Qry.ParamByName('Chave').AsInteger := Chave;
    Qry.ParamByName('Pendente').AsBoolean := true;
    Qry.Open;
    Qry.First;
    while (not Qry.Eof) do
    begin
        Lista := Lista+Qry.fieldbyname('ChaveUsuario').AsString+separador+
            Qry.fieldbyname('Nome').AsString+separador+
            Qry.fieldbyname('Email').AsString+separador;
        Qry.Next;
    end;
    if (Lista<>") then

DtModule.Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocol
o('017'+Lista));
    freeandnil(Qry);
end;

procedure TDtModule.EncaminhaMsg(Msg: string; Cliente: Pointer);
var Lista: TStringList;
    ChaveContato: integer;
    Indice: integer;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    ChaveContato := strtoint(Lista[1]);
    Indice := ChavesUsuarios.IndexOf(inttostr(ChaveContato));

```

```

DtModule.Server.Socket.Connections[LocalizaCliente(IDConexoes[Indice])].SendText(Protocolo('027'+Msg));
    freeandnil(Lista);
end;

```

```

procedure TDtModule.EnviaDadosGerais(Msg: string; Cliente: Pointer);
var Txt: string;
    Qtd: integer;
begin
    Qtd := (length(Servidores) div 8);
    if ((length(Servidores) mod 8)>0) then
        inc(Qtd);
    Txt := '036'+inttostr(Qtd);

```

```

DtModule.Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo(Txt));
end;

```

```

procedure TDtModule.EnviaPaginaServidores(Msg: string; Cliente: Pointer);
var Pagina: integer;
    Indice: integer;
    x: integer;
    Txt: string;
begin
    Pagina:= strtoint(Msg);
    Indice := ((Pagina-1)*8);
    for x := Indice to Indice+7 do
        begin
            if (x>=length(Servidores)) then
                break;

```

```

Txt := '031'+Msg+separador+inttostr(x mod 8)+separador;
Txt := Txt+inttostr(Servidores[x].ChaveServidor)+separador;
Txt := Txt+Servidores[x].Titulo+separador;
Txt := Txt+Servidores[x].Descricao+separador;
Txt := Txt+inttostr(Servidores[x].NumMaxUsuarios)+separador;
Txt := Txt+Servidores[x].HostName+separador;
Txt := Txt+Servidores[x].NomeArquivoMapa+separador;
Txt := Txt+Servidores[x].NomeArquivoTextura;

```

```

DtModule.Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocol
o(Txt));
end;
end;

```

```

procedure TDtModule.EfetuaLoginMapa(Msg: string; Cliente: Pointer);

```

```

var Qry: TZQuery;
    Lista: TStringList;
    Index: integer;

```

```

begin

```

```

    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select * from Servidor where Login=:Login');
    Qry.ParamByName('Login').AsString := Lista[0];
    Qry.Open;
    if (Qry.RecordCount=0) then
    begin

```

```

        Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('042Logi
n Inválido!'))
    end;
end;

```



```

end
else
begin
  if (Qry.FieldName('Senha').AsString=Lista[1]) then
  begin
    Index := IDConexoes.IndexOf(Cliente);
    TiposUsuarios.Strings[Index] := 'Servidor de Mapa';
    fsPrincipal.Grid.Cells[3,Index+1] := 'Servidor de Mapa';
    Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('041'

+Qry.fieldbyname('ChaveServidor').AsString+separador

+Qry.fieldbyname('Titulo').AsString+separador

+Qry.fieldbyname('Descricao').AsString+separador

+Qry.fieldbyname('Capacidade').AsString+separador

+Qry.fieldbyname('Hostname').AsString+separador

+Qry.fieldbyname('NomeArquivoMapa').AsString+separador

+Qry.fieldbyname('NomeArquivoTextura').AsString));
    end
    else

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('042Sen
ha incorreta!'));
    end;
    freeandnil(Qry);
end;

```

```

procedure TDtModule.KickarUsuario(Index: integer; Motivo: string);
begin
  Server.Socket.Connections[Index].SendText(Protocolo('048'+Motivo));
end;

```

```

procedure TDtModule.AdicionaContatoPelaChave(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Index: integer;
    Chave: integer;
    ChaveUsuario: integer;
    Contato: string;
    Lista: string;
begin
  ChaveUsuario := strtoint(ChavesUsuarios[LocalizaCliente(Cliente)]);
  Chave := strtoint(Msg);
  Qry := TZQuery.Create(nil);
  Qry.Connection := dtModule.Connection;
  Qry.SQL.Clear;
  Qry.SQL.Add('Select ChaveAmigo, Pendente from UsuarioContato');
  Qry.SQL.Add('where (ChaveUsuario=:ChaveUsuario and
ChaveAmigo=:ChaveAmigo)');
  Qry.SQL.Add('or (ChaveUsuario=:ChaveAmigo and
ChaveAmigo=:ChaveUsuario)');
  Qry.ParamByName('ChaveUsuario').AsInteger := ChaveUsuario;
  Qry.ParamByName('ChaveAmigo').AsInteger := Chave;
  Qry.Open;
  if (Qry.RecordCount=0) then
  begin
    Qry.SQL.Clear;

```

```

    Qry.SQL.Add('Insert into UsuarioContato
(ChaveUsuario,ChaveAmigo,Pendente)');
    Qry.SQL.Add('values (:ChaveUsuario,:ChaveAmigo,:Pendente)');
    Qry.ParamByName('ChaveUsuario').AsInteger := ChaveUsuario;
    Qry.ParamByName('ChaveAmigo').AsInteger := Chave;
    Qry.ParamByName('Pendente').AsBoolean := true;
    Qry.ExecSQL;
    Qry.SQL.Clear;
    Qry.SQL.Add('Select ChaveUsuario, Nome, Email from Usuario where
ChaveUsuario=:Chave');
    Qry.ParamByName('Chave').AsInteger := ChaveUsuario;
    Qry.Open;
    Lista := '';
    Lista := Lista+Qry.fieldbyname('ChaveUsuario').AsString+separador+
        Qry.fieldbyname('Nome').AsString+separador+
        Qry.fieldbyname('Email').AsString+separador;
    Index := ChavesUsuarios.IndexOf(inttostr(Chave));
    if (Index>-1) then
        DtModule.Server.Socket.Connections[Index].SendText(Protocolo('017'+Lista));
    end
    else if (Qry.FieldByName('ChaveAmigo').AsInteger=ChaveUsuario) and
(Qry.FieldByName('Pendente').AsBoolean) then
    begin
        Qry.SQL.Clear;
        Qry.SQL.Add('Update UsuarioContato set Pendente=:Pendente');
        Qry.SQL.Add('where ChaveUsuario=:ChaveUsuario and
ChaveAmigo=:ChaveAmigo');
        Qry.ParamByName('ChaveUsuario').AsInteger := Chave;
        Qry.ParamByName('ChaveAmigo').AsInteger := ChaveUsuario;
        Qry.ParamByName('Pendente').AsBoolean := false;
        Qry.ExecSQL;

```

```

Qry.SQL.Clear;
Qry.SQL.Add('Select Apelido, StatusUsuario, Sexo, Pais, Email from Usuario');
Qry.SQL.Add('where ChaveUsuario=:Chave');
Qry.ParamByName('Chave').AsInteger := Chave;
Qry.Open;
Contato := inttostr(Chave)+Separador+
    Qry.fieldbyname('Apelido').AsString+Separador+
    Qry.fieldbyname('StatusUsuario').AsString+Separador+
    Qry.fieldbyname('Sexo').AsString+Separador+
    Qry.fieldbyname('Pais').AsString+Separador+
    Qry.fieldbyname('Email').AsString+Separador;

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('014'+Co
ntato));

Qry.SQL.Clear;
Qry.SQL.Add('Select Apelido, StatusUsuario, Sexo, Pais, Email from Usuario');
Qry.SQL.Add('where ChaveUsuario=:Chave');
Qry.ParamByName('Chave').AsInteger := ChaveUsuario;
Qry.Open;
Contato := inttostr(ChaveUsuario)+Separador+
    Qry.fieldbyname('Apelido').AsString+Separador+
    Qry.fieldbyname('StatusUsuario').AsString+Separador+
    Qry.fieldbyname('Sexo').AsString+Separador+
    Qry.fieldbyname('Pais').AsString+Separador+
    Qry.fieldbyname('Email').AsString+Separador;
Index := ChavesUsuarios.IndexOf(inttostr(Chave));
if (Index>-1) then
    Server.Socket.Connections[Index].SendText(Protocolo('014'+Contato));
end;
freeandnil(Qry);
end;

```

```

procedure TDtModule.EncaminhaNudge(Msg: string; Cliente: Pointer);
var ChaveContato: integer;
    Indice: integer;
    Lista: TStringList;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    ChaveContato := strtoint(Lista[1]);
    Indice := ChavesUsuarios.IndexOf(inttostr(ChaveContato));

    DtModule.Server.Socket.Connections[LocalizaCliente(IDConexoes[Indice])].SendText(Protocolo('051'+Lista[0]));
    freeandnil(Lista);
end;

procedure TDtModule.RespondePing(Msg: string; Cliente: Pointer);
begin

    DtModule.Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('054'));
end;

procedure TDtModule.EnviaDadosContato(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Lista: TStringList;
    chave_usuario: integer;
    email: string;
    nome: string;
    apelido: string;
    nascimento: tdatetime;

```

```

sexo: integer;
estado_civil: integer;
cidade: string;
estado: string;
pais: string;
telefone: string;
celular: string;
chave_modelo: integer;
nome_arquivo: string;
begin
  Lista := TStringList.Create;
  SeparaString(Msg, Lista);
  Qry := TZQuery.Create(nil);
  Qry.Connection := Connection;
  Qry.SQL.Add('Select * from Usuario U left join Modelo M on
U.ChaveModelo=M.ChaveModelo');
  Qry.SQL.Add('where ChaveUsuario=:usuario');
  Qry.ParamByName('usuario').AsInteger := strtoint(Lista.Strings[0]);
  Qry.Open;
  if (Qry.RecordCount>0) then
  begin
    chave_usuario := Qry.fieldbyname('ChaveUsuario').AsInteger;
    email := Qry.fieldbyname('Email').AsString;
    nome := Qry.fieldbyname('Nome').AsString;
    apelido := Qry.fieldbyname('Apelido').AsString;
    nascimento := Qry.fieldbyname('Nascimento').AsDateTime;
    sexo := Qry.fieldbyname('Sexo').AsInteger;
    estado_civil := Qry.fieldbyname('EstadoCivil').AsInteger;
    cidade := Qry.fieldbyname('Cidade').AsString;
    estado := Qry.fieldbyname('Estado').AsString;
    pais := Qry.fieldbyname('Pais').AsString;
  end
end

```

```

telefone := Qry.fieldbyname('Telefone').AsString;
celular := Qry.fieldbyname('Celular').AsString;
chave_modelo := Qry.fieldbyname('ChaveModelo').AsInteger;
nome_arquivo := Qry.fieldbyname('NomeArquivo').AsString;

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('018'+inttostr(chave_usuario)+separador+
nome+separador+
apelido+separador+
datetostr(nascimento)+separador+
inttostr(sexo)+separador+
inttostr(estados_civil)+separador+
cidade+separador+
estado+separador+
pais+separador+
telefone+separador+
celular+separador+
'4'+separador+
inttostr(chave_modelo)+separador+
nome_arquivo+separador+
email));

end;
freeandnil(Qry);
freeandnil(Lista);
end;

procedure TDtModule.RegistraEnvioDeArquivo(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Lista: TStringList;
    Index: integer;
    Info: string;

```

```

begin
  Lista := TStringList.Create;
  SeparaString(Msg, Lista);
  Qry := TZQuery.Create(nil);
  Qry.Connection := Connection;
  Qry.SQL.Add('Insert into ArquivosEnviados (ChaveRemetente,
ChaveDestinatario, NomeArquivo, Data)');
  Qry.SQL.Add('values (:Remetente, :Destinatario, :NomeArquivo, :Data)');
  Qry.ParamByName('Remetente').AsInteger := strtoint(Lista[0]);
  Qry.ParamByName('Destinatario').AsInteger := strtoint(Lista[1]);
  Qry.ParamByName('NomeArquivo').AsString := Lista[2];
  Qry.ParamByName('Data').AsDate := now;
  Qry.ExecSQL;
  Index := ChavesUsuarios.IndexOf(Lista[1]);
  if (Index>-1) then
    begin
      Qry.SQL.Clear;
      Qry.SQL.Add('Select Nome, Email from Usuario where ChaveUsuario=:Chave');
      Qry.ParamByName('Chave').AsInteger := strtoint(Lista[0]);
      Qry.Open;
      Info :=
Lista[0]+separador+Qry.fieldbyname('Nome').AsString+separador+Qry.fieldbynam
e('Email').AsString+separador+Lista[2];
      Index := ChavesUsuarios.IndexOf(Lista[1]);
      if (Index>-1) then
        Server.Socket.Connections[Index].SendText(Protocolo('060'+Info));
      end;
      freeandnil(Qry);
      freeandnil(Lista);
    end;
end;

```



```

procedure TDtModule.EnviaListaArquivos(Msg: string; Cliente: Pointer);
var Qry: TZQuery;
    Index: integer;
    Chave: integer;
    Lista: string;
begin
    try
        Lista := '';
        Index := LocalizaCliente(Cliente);
        Chave := strtoint(ChavesUsuarios[Index]);
        Qry := TZQuery.Create(nil);
        Qry.Connection := DtModule.Connection;
        Qry.SQL.Add('Select U.ChaveUsuario, U.Nome, U.Email, AE.NomeArquivo');
        Qry.SQL.Add('from Usuario U, ArquivosEnviados AE');
        Qry.SQL.Add('where AE.ChaveRemetente=U.ChaveUsuario');
        Qry.SQL.Add('and AE.ChaveDestinatario=:Destinatario');
        Qry.ParamByName('Destinatario').AsInteger := Chave;
        Qry.Open;
        Qry.First;
        while (not Qry.Eof) do
            begin
                Lista := Lista+Qry.fieldbyname('ChaveUsuario').AsString+separador+
                    Qry.fieldbyname('Nome').AsString+separador+
                    Qry.fieldbyname('Email').AsString+separador+
                    Qry.fieldbyname('NomeArquivo').AsString+separador;
                Qry.Next;
            end;
        if (Lista<>'') then

DtModule.Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocol
o('060'+Lista));

```

```

finally
    freeandnil(Qry);
end;
end;

```

```

procedure TDtModule.ExclueRegistroArquivoEnviado(Msg: string; Cliente:
Pointer);
var Qry: TZQuery;
begin
    Qry := TZQuery.Create(nil);
    Qry.Connection := connection;
    Qry.SQL.Add('Delete from ArquivosEnviados where NomeArquivo=:Arquivo');
    Qry.ParamByName('Arquivo').AsString := Msg;
    Qry.ExecSQL;
    freeandnil(Qry);
    //EnviaListaArquivos("", Cliente);
end;

```

```

// -----
//  Fim das mensagens recebidas dos clientes
// -----

```

```

// -----
//  Outras funções
// -----

```

```

procedure TDtModule.PreencheListaAvatares;
var Qry: TZQuery;
begin
    Avatares := "";
    Qry := TZQuery.Create(nil);

```

```

try
  Qry.Connection := DtModule.Connection;
  Qry.SQL.Add('Select * from Modelo order by NomeArquivo');
  Qry.Open;
  Qry.First;
  if (Qry.RecordCount>0) then
    begin
      repeat
        Avatares := Avatares+Qry.FieldName('NomeArquivo').AsString+separador
          +Qry.FieldName('ChaveModelo').AsString+separador;
        Qry.Next;
      until (Qry.Eof);
    end;
  finally
    freeandnil(Qry);
  end;
end;

```

```

procedure TDtModule.PreencheListaServidores;
var Qry: TZQuery;
    x: integer;
begin
  setlength(Servidores,0);
  Qry := TZQuery.Create(nil);
  try
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select * from Servidor order by ChaveServidor');
    Qry.Open;
    Qry.First;
    if (Qry.RecordCount>0) then
      begin

```

```

x := 0;
setlength(Servidores, Qry.RecordCount);
repeat
  Servidores[x] := TMapa.Create;
  Servidores[x].ChaveServidor := Qry.fieldbyname('ChaveServidor').AsInteger;
  Servidores[x].Titulo := Qry.fieldbyname('Titulo').AsString;
  Servidores[x].Descricao := Qry.fieldbyname('Descricao').AsString;
  Servidores[x].NumMaxUsuarios := Qry.fieldbyname('Capacidade').AsInteger;
  Servidores[x].HostName := Qry.fieldbyname('Hostname').AsString;
  Servidores[x].NomeArquivoMapa :=
Qry.fieldbyname('NomeArquivoMapa').AsString;
  Servidores[x].NomeArquivoTextura :=
Qry.fieldbyname('NomeArquivoTextura').AsString;
  Qry.Next;
  inc(x);
until (Qry.Eof);
end;
finally
  freeandnil(Qry);
end;
end;

function TDtModule.LocalizaCliente(Cliente: Pointer): integer;
begin
  result := IDConexoes.IndexOf(Cliente);
end;

procedure TDtModule.SetarTodosUsuariosOffline;
var Qry: TZQuery;
begin
  Qry := TZQuery.Create(nil);

```

```

try
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Update Usuario set StatusUsuario=:Status');
    Qry.ParamByName('Status').AsInteger := -1;
    Qry.ExecSQL;
finally
    freeandnil(Qry);
end;
end;

procedure TDtModule.ListarAmigosUsuario(ChaveUsuario: integer; var Qry:
TZQuery);
begin
    Qry.SQL.Clear;
    Qry.SQL.Add('(Select ChaveAmigo as Chave from UsuarioContato');
    Qry.SQL.Add('where ChaveUsuario=:Usuario)');
    Qry.SQL.Add('Union');
    Qry.SQL.Add('(Select ChaveUsuario as Chave from UsuarioContato');
    Qry.SQL.Add('where ChaveAmigo=:Usuario)');
    Qry.SQL.Add('order by Chave');
    Qry.ParamByName('Usuario').AsInteger := ChaveUsuario;
    Qry.Open;
end;

procedure TDtModule.SetarStatusUsuario(ChaveUsuario: integer; Status: integer);
var Qry: TZQuery;
    Chave: integer;
    Index: integer;
begin
    Qry := TZQuery.Create(nil);
    try

```

```

Qry.Connection := DtModule.Connection;
Qry.SQL.Add('Update Usuario set StatusUsuario=:Status');
Qry.SQL.Add('where ChaveUsuario=:Chave');
Qry.ParamByName('Status').AsInteger := Status;
Qry.ParamByName('Chave').AsInteger := ChaveUsuario;
Qry.ExecSQL;

ListarAmigosUsuario(ChaveUsuario, Qry);
Qry.First;
while not Qry.Eof do
begin
    Chave := Qry.fieldbyname('Chave').AsInteger;
    Index := ChavesUsuarios.IndexOf(inttostr(Chave));
    if (Index<>-1) then

DtModule.Server.Socket.Connections[Index].SendText(Protocolo('020'+inttostr(Ch
aveUsuario)+separador+inttostr(Status)));
        Qry.Next;
    end;
finally
    freeandnil(Qry);
end;
end;

// -----
//  Fim das outras funções
// -----

end.

```

```
unit uDtModuleClient;
```

```
interface
```

```
uses
```

```
    SysUtils, Classes, ScktComp, uFuncoes, Dialogs, Forms, WinSkinData,  
    uConfiguracoes;
```

```
type
```

```
    TDtModuleClient = class(TDataModule)
```

```
        Client: TClientSocket;
```

```
        ClientMap: TClientSocket;
```

```
        Server: TServerSocket;
```

```
        SkinData: TSkinData;
```

```
        procedure ServerClientConnect(Sender: TObject; Socket: TCustomWinSocket);
```

```
        procedure ServerClientDisconnect(Sender: TObject; Socket:
```

```
TCustomWinSocket);
```

```
        procedure DataModuleCreate(Sender: TObject);
```

```
        procedure DataModuleDestroy(Sender: TObject);
```

```
        procedure ClientError(Sender: TObject; Socket: TCustomWinSocket;
```

```
ErrorEvent: TErrorEvent; var ErrorCode: Integer);
```

```
        procedure ClientDisconnect(Sender: TObject; Socket: TCustomWinSocket);
```

```
        procedure ServerClientError(Sender: TObject; Socket: TCustomWinSocket;
```

```
            ErrorEvent: TErrorEvent; var ErrorCode: Integer);
```

```
    private
```

```
        { Private declarations }
```

```
    public
```

```
        IDConexoes: TList; //Gerenciador de conexões
```

```
        ChavesUsuarios: TStringList;
```

```
        EstadosUsuarios: TStringList;
```

```
        ChavesMsgsEmEspera: TStringList;
```

```

    MsgsEmEspera: TStringList;
    { Public declarations }
end;

var
    DtModuleClient: TDtModuleClient;

implementation

{$R *.dfm}

procedure TDtModuleClient.ServerClientConnect(Sender: TObject; Socket:
TCustomWinSocket);
var IDsocket: ^byte;
begin
    New(IDSocket);
    Socket.Data := IDSocket;
    IDConexoes.Add(Socket.data);
end;

procedure TDtModuleClient.ServerClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
var Index: integer;
    Chave: integer;
begin
    Index := IDConexoes.IndexOf(Socket.Data);
    Chave := strtoint(ChavesUsuarios[Index]);
    ChavesUsuarios.Delete(Index);
    EstadosUsuarios.Delete(Index);
    IDConexoes.Remove(Socket.Data);
    Dispose(Socket.Data);

```


end;

procedure TDtModuleClient.DataModuleCreate(Sender: TObject);

begin

Client.Host := configuracoes.ServerHost;

IDConexoes := TList.Create;

ChavesUsuarios := TStringList.Create();

EstadosUsuarios := TStringList.Create();

MsgsEmEspera := TStringList.Create();

ChavesMsgsEmEspera := TStringList.Create();

IDConexoes.Clear;

ChavesUsuarios.Clear;

EstadosUsuarios.Clear;

MsgsEmEspera.Clear;

ChavesMsgsEmEspera.Clear;

SkinData.Active := configuracoes.Skin;

end;

procedure TDtModuleClient.DataModuleDestroy(Sender: TObject);

begin

freeandnil(IDConexoes);

freeandnil(ChavesUsuarios);

freeandnil(EstadosUsuarios);

freeandnil(MsgsEmEspera);

freeandnil(ChavesMsgsEmEspera);

end;

procedure TDtModuleClient.ClientError(Sender: TObject; Socket:

TCustomWinSocket; ErrorEvent: TErrorEvent;

```

    var ErrorCode: Integer);
begin
    if (ErrorCode=10061) then
    begin
        showmessage('Não foi possível se conectar ao Servidor 3D Blah. Por favor,
tente novamente mais tarde!');
        ErrorCode := 0;
        application.Terminate;
    end
    else
    begin
        showmessage('Erro '+inttostr(ErrorCode));
        ErrorCode := 0;
    end;
end;

```

```

procedure TDtModuleClient.ClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
begin
    showmessage('A conexão com o servidor foi perdida. A aplicação será
fechada!');
    application.Terminate;
end;

```

```

procedure TDtModuleClient.ServerClientError(Sender: TObject; Socket:
TCustomWinSocket; ErrorEvent: TErrorEvent;
    var ErrorCode: Integer);
begin
    ErrorCode := 0;
end;

```

end.

unit uDtModuleDialog;

interface

uses

SysUtils, Classes, ScktComp, Forms, uFuncoes, WinSkinData;

type

TDtModuleDialog = class(TDataModule)

Client: TClientSocket;

SkinData: TSkinData;

procedure DataModuleCreate(Sender: TObject);

procedure ClientDisconnect(Sender: TObject; Socket: TCustomWinSocket);

procedure ClientConnect(Sender: TObject; Socket: TCustomWinSocket);

procedure ClientError(Sender: TObject; Socket: TCustomWinSocket;

ErrorEvent: TErrorEvent; var ErrorCode: Integer);

private

{ Private declarations }

public

{ Public declarations }

end;

var

DtModuleDialog: TDtModuleDialog;

implementation

uses ufDialogo;

```
{$R *.dfm}
```

```
procedure TDtModuleDialog.DataModuleCreate(Sender: TObject);
begin
  if (ParamStr(1)='S') then
    SkinData.Active := true
  else
    SkinData.Active := false;
end;
```

```
procedure TDtModuleDialog.ClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
begin
  Application.Terminate;
end;
```

```
procedure TDtModuleDialog.ClientConnect(Sender: TObject; Socket:
TCustomWinSocket);
begin
```

```
DtModuleDialog.Client.Socket.SendText(Protocolo('021'+inttostr(fDialogo.Contato.
chave_usuario)));
```

```
DtModuleDialog.Client.Socket.SendText(Protocolo('057'+inttostr(fDialogo.Contato.
chave_usuario)));
end;
```

```
procedure TDtModuleDialog.ClientError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
```

```
begin
```

```
    ErrorCode := 0;
```

```
end;
```

```
end.
```

```
unit uDtModuleMap;
```

```
interface
```

```
uses
```

```
    SysUtils, Classes, ScktComp, uTolderlundStringGrid, uFuncoes, uConstantes,  
    Dialogs, Forms, uConfiguracoes, Direct3D8, TrueVision3D;
```

```
type
```

```
    TDtModuleMap = class(TDataModule)
```

```
        Server: TServerSocket;
```

```
        Client: TClientSocket;
```

```
        procedure ServerClientConnect(Sender: TObject; Socket: TCustomWinSocket);
```

```
        procedure ServerClientDisconnect(Sender: TObject; Socket:
```

```
TCustomWinSocket);
```

```
        procedure DataModuleCreate(Sender: TObject);
```

```
        procedure DataModuleDestroy(Sender: TObject);
```

```
        procedure ServerClientRead(Sender: TObject; Socket: TCustomWinSocket);
```

```
        procedure ClientError(Sender: TObject; Socket: TCustomWinSocket;
```

```
ErrorEvent: TErrorEvent; var ErrorCode: Integer);
```

```
        procedure ClientDisconnect(Sender: TObject; Socket: TCustomWinSocket);
```

```
        procedure ServerClientError(Sender: TObject; Socket: TCustomWinSocket;
```

```
            ErrorEvent: TErrorEvent; var ErrorCode: Integer);
```

```
    private
```

```

    { Private declarations }
public
    IDConexoes: TList; //Gerenciador de conexões
    ChavesUsuarios: TStringList;
    ArquivosModelos: TStringList;
    Apelidos: TStringList;
    PosInicio: D3DVector;
    PosicoesX: TStringList;
    PosicoesY: TStringList;
    PosicoesZ: TStringList;
    RotacoesY: TStringList;
    procedure InterpretaMsg(Codigo: integer; Msg: string; Cliente: Pointer);
    function LocalizaCliente(Cliente: Pointer): integer;
    procedure NovoClienteConectou(Msg: string; Cliente: Pointer);
    procedure EnviaListaUsuarios(Msg: string; Cliente: Pointer);
    procedure EncaminhaPosicaoUsuario(Msg: string; Cliente: Pointer);
    procedure UsuarioConectou(ChaveUsuario: string);
    procedure UsuarioDesconectou(ChaveUsuario: string);
    procedure SeDesconectaDoServidor(Msg: string);
    procedure KickarUsuario(Index: integer; Motivo: string);
    { Public declarations }
end;

var
    DtModuleMap: TDtModuleMap;

implementation

uses ufmPrincipal;

{$R *.dfm}

```

```

// -----
//  Eventos dos componentes
// -----

procedure TDtModuleMap.ServerClientConnect(Sender: TObject; Socket:
TCustomWinSocket);
var IDsocket: ^byte;
begin
    New(IDSocket);
    Socket.Data := IDSocket;
    IDConexoes.Add(Socket.data);
    ChavesUsuarios.Add('0');
    ArquivosModelos.Add('0');
    Apelidos.Add('');
    PosicoesX.Add(floattostr(PosInicio.x));
    PosicoesY.Add(floattostr(PosInicio.y));
    PosicoesZ.Add(floattostr(PosInicio.z));
    RotacoesY.Add('0');
    inc(fmPrincipal.Mapa.NumUsuarios);
    fmPrincipal.AtualizaNumeroUsuarios();
    fmPrincipal.Grid.RowCount := Server.Socket.ActiveConnections+1;
    fmPrincipal.Grid.Cells[0,Server.Socket.ActiveConnections] :=
Socket.RemoteHost;
    fmPrincipal.Grid.Cells[2,Server.Socket.ActiveConnections] := datetostr(now)+' -
'+timetostr(now);
end;

procedure TDtModuleMap.ServerClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
var Index: integer;

```

```

begin
    Index := IDConexoes.IndexOf(Socket.Data);
    UsuarioDesconectou(ChavesUsuarios[Index]);
    ChavesUsuarios.Delete(Index);
    ArquivosModelos.Delete(Index);
    Apelidos.Delete(Index);
    PosicoesX.Delete(Index);
    PosicoesY.Delete(Index);
    PosicoesZ.Delete(Index);
    RotacoesY.Delete(Index);
    TStringGridHack(fmPrincipal.Grid).DeleteRow(Index+1);
    IDConexoes.Remove(Socket.Data);
    Dispose(Socket.Data);
    dec(fmPrincipal.Mapa.NumUsuarios);
    fmPrincipal.AtualizaNumeroUsuarios();
end;

procedure TDtModuleMap.DataModuleCreate(Sender: TObject);
begin
    Client.Host := configuracoes.ServerHost;
    IDConexoes := TList.Create;
    IDConexoes.Clear;
    ChavesUsuarios := TStringList.Create();
    ChavesUsuarios.Clear;
    ArquivosModelos := TStringList.Create();
    ArquivosModelos.Clear;
    Apelidos := TStringList.Create();
    Apelidos.Clear;
    PosicoesX := TStringList.Create();
    PosicoesX.Clear;
    PosicoesY := TStringList.Create();

```



```

PosicoesY.Clear;
PosicoesZ := TStringList.Create();
PosicoesZ.Clear;
RotacoesY := TStringList.Create();
RotacoesY.Clear;
end;

```

```

procedure TDtModuleMap.DataModuleDestroy(Sender: TObject);
begin
    freeandnil(IDConexoes);
    freeandnil(ChavesUsuarios);
    freeandnil(ArquivosModelos);
    freeandnil(Apelidos);
    freeandnil(PosicoesX);
    freeandnil(PosicoesY);
    freeandnil(PosicoesZ);
    freeandnil(RotacoesY);
end;

```

```

procedure TDtModuleMap.ClientError(Sender: TObject; Socket:
TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);
begin
    if (ErrorCode=10061) then
    begin
        showmessage('Não foi possível se conectar ao Servidor 3D Blah. Por favor,
tente novamente mais tarde!');
        ErrorCode := 0;
        application.Terminate;
    end;
end;

```

```

procedure TDtModuleMap.ClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
begin
  if (fmPrincipal<>nil) then
  begin
    showmessage('A conexão com o servidor foi perdida. A aplicação será
fechada!');
    application.Terminate;
  end;
end;

```

```

procedure TDtModuleMap.ServerClientError(Sender: TObject; Socket:
TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  ErrorCode := 0;
end;

```

```

// -----
// Fim dos eventos dos componentes
// -----

```

```

// -----
// Trata as mensagens recebidas dos clientes
// -----

```

```

procedure TDtModuleMap.ServerClientRead(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Cliente: Pointer;

```

```

    Lista: TStringList;
    x: integer;
begin
    Cliente := Socket.Data;
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x],Codigo,Msg);
                InterpretaMsg(Codigo,Msg,Cliente);
            end;
        finally
            freeandnil(Lista);
        end;
    end;
end;

```

```

procedure TDtModuleMap.InterpretaMsg(Codigo: integer; Msg: string; Cliente:
Pointer);
begin
    if (Codigo=032) then
        NovoClienteConectou(Msg, Cliente)
    else if (Codigo=037) then
        EnviaListaUsuarios(Msg, Cliente)
    else if (Codigo=045) then
        EncaminhaPosicaoUsuario(Msg, Cliente);
    end;
end;

```

```

procedure TDtModuleMap.NovoClienteConectou(Msg: string; Cliente: Pointer);
var Lista: TStringList;
begin

```

```

Lista := TStringList.Create;
try
  SeparaString(Msg, Lista);
  if (strtoint(Lista[3]) <> fmPrincipal.Mapa.ChaveServidor) then
    begin

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('034Servidor inativo no momento. Por favor, selecione outro ou tente novamente mais tarde!'));

      end
    else if
      (FileSize(extractfilepath(application.ExeName)+'Maps\' + fmPrincipal.Mapa.NomeArquivoMapa) <> strtoint(Lista[4])) then
        begin

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('034O mapa encontrado em seu computador é diferente do mapa usado pelo servidor. Por favor, baixe a versão mais recente do mapa em www.3dblah.com.br!'));

          end
        else if (fmPrincipal.Mapa.NumUsuarios > fmPrincipal.Mapa.NumMaxUsuarios)
        then
          begin

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('034Este servidor está cheio. Por favor, selecione outro ou tente novamente mais tarde!'));

            end
          else
            begin
              ChavesUsuarios.Strings[LocalizaCliente(Cliente)] := Lista[0];
              ArquivosModelos.Strings[LocalizaCliente(Cliente)] := Lista[1];
              Apelidos.Strings[LocalizaCliente(Cliente)] := Lista[2];
            end
          end
        end
      end
    end
  end

```

```

Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo('033'));
    UsuarioConectou(Lista[0]);
end;
finally
    freeandnil(Lista);
end;
end;

```

```

procedure TDtModuleMap.EnviaListaUsuarios(Msg: string; Cliente: Pointer);
var x: integer;
    Txt: string;
begin
    Txt := '038';
    for x := 0 to ChavesUsuarios.Count-1 do
        begin
            if (x<>LocalizaCliente(Cliente)) then
                Txt := Txt+ChavesUsuarios[x]+separador
                    +ArquivosModelos[x]+separador
                    +Apelidos[x]+separador
                    +PosicoesX[x]+separador
                    +PosicoesY[x]+separador
                    +PosicoesZ[x]+separador
                    +RotacoesY[x]+separador;
            end;
        Server.Socket.Connections[LocalizaCliente(Cliente)].SendText(Protocolo(Txt));
    end;

```

```

procedure TDtModuleMap.EncaminhaPosicaoUsuario(Msg: string; Cliente:
Pointer);
var Lista: TStringList;

```

```

    x: integer;
begin
    Lista := TStringList.Create;
    try
        SeparaString(Msg, Lista);
        PosicoesX[LocalizaCliente(Cliente)] := Lista[0];
        PosicoesY[LocalizaCliente(Cliente)] := Lista[1];
        PosicoesZ[LocalizaCliente(Cliente)] := Lista[2];
        RotacoesY[LocalizaCliente(Cliente)] := Lista[3];
        for x := 0 to ChavesUsuarios.Count-1 do
            begin
                if (x<>LocalizaCliente(Cliente)) then

Server.Socket.Connections[x].SendText(Protocolo('046'+ChavesUsuarios[Localiza
Cliente(Cliente)]+separador
                                +Lista[0]+separador
                                +Lista[1]+separador
                                +Lista[2]+separador
                                +Lista[3]));

                end;
            finally
                freeandnil(Lista);
            end;
        end;

// -----
//  Fim das mensagens recebidas dos clientes
// -----

// -----
//  Outras funções

```

```
// -----
```

```
function TDtModuleMap.LocalizaCliente(Cliente: Pointer): integer;
begin
    result := IDConexoes.IndexOf(Cliente);
end;
```

```
procedure TDtModuleMap.UsuarioDesconectou(ChaveUsuario: string);
var x: integer;
begin
    for x := 0 to ChavesUsuarios.Count-1 do
    begin
        try
            Server.Socket.Connections[x].SendText(Protocolo('044'+ChaveUsuario));
        except
        end;
    end;
end;
```

```
procedure TDtModuleMap.SeDesconectaDoServidor(Msg: string);
begin
    if (length(Msg)>0) then
        showmessage('Você foi desconectado do Servidor pelo administrador! Motivo: '+Msg)
    else
        showmessage('Você foi desconectado do Servidor pelo administrador!');
        Client.Close;
end;
```

```
procedure TDtModuleMap.UsuarioConectou(ChaveUsuario: string);
var x, y: integer;
```

```

begin
  for y := 0 to ChavesUsuarios.Count-1 do
    if (ChavesUsuarios[y]=Chaveusuario) then
      break;
    if (ChavesUsuarios[y]=Chaveusuario) then
      for x := 0 to ChavesUsuarios.Count-1 do
        begin
          if (x<>y) then

Server.Socket.Connections[x].SendText(Protocolo('043'+ChaveUsuario+separador
                                     +ArquivosModelos[y]+separador
                                     +Apelidos[y]+separador
                                     +PosicoesX[y]+separador
                                     +PosicoesY[y]+separador
                                     +PosicoesZ[y]+separador
                                     +RotacoesY[y]));

          end;
        end;

procedure TDtModuleMap.KickarUsuario(Index: integer; Motivo: string);
begin
  Server.Socket.Connections[Index].SendText(Protocolo('058'+Motivo));
end;

// -----
//  Fim das outras funções
// -----

end.

```



```
unit uEnviaArquivo;
```

```
interface
```

```
uses
```

```
Classes, IdComponent, IdTCPConnection, IdTCPClient, IdFTP, uFTP, SysUtils,  
uFuncoes, ExtCtrls, Graphics, Dialogs, ComCtrls;
```

```
type
```

```
TEnviaArquivo = class(TThread)
```

```
private
```

```
Diretorios: TStringList;
```

```
Origem: string;
```

```
Progresso: TProgressBar;
```

```
{ Private declarations }
```

```
public
```

```
IdFTP: TIdFTP;
```

```
Arquivo: string;
```

```
constructor Criar(ArquivoOrigem: string; Barra: TProgressBar);
```

```
destructor Destroy; override;
```

```
procedure IdFTPWork(ASender: TObject; AWorkMode: TWorkMode;
```

```
AWorkCount: Integer);
```

```
protected
```

```
procedure Execute; override;
```

```
end;
```

```
implementation
```

```
{ TBaixaArquivo }
```

```
constructor TEnviaArquivo.Criar(ArquivoOrigem: string; Barra: TProgressBar);
```

```

begin
  Create(true);
  IdFTP := TIdFTP.Create(nil);
  Diretorios := TStringList.Create;
  FreeOnTerminate := true;
  Origem := ArquivoOrigem;
  Progresso := Barra;
  IdFTP.OnWork := IdFTPWork;
end;

destructor TEnviaArquivo.Destroy;
begin
  freeandnil(IdFTP);
  freeandnil(Diretorios);
  inherited;
end;

procedure TEnviaArquivo.Execute;
var x: integer;
begin
  try
    try
      IdFTP.Username := FTP.FTPUser;
      IdFTP.Password := FTP.FTPPass;
      IdFTP.Host := FTP.FTPHost;
      IdFTP.Connect;
      GetDirList(FTP.FTPDir, Diretorios);
      for x := 0 to Diretorios.Count-1 do
        IdFTP.ChangeDir(Diretorios[x]);
        IdFTP.ChangeDir(FTP.FTPFileDir);
      x := 0;
    end;
  end;
end;

```

```

repeat
  inc(x);
  Arquivo := '('+inttostr(x)+')'+extractfilename(Origem);
until (IdFTP.Size(Arquivo)=-1);
Progresso.Max := FileSize(Origem);
if (FileExists(Arquivo)) then
  DeleteFile(Arquivo);
IdFTP.Put(Origem, Arquivo);
except
  showmessage('Erro na transferência do arquivo!');
end;
IdFTP.Disconnect
finally
end;
end;

procedure TEnviaArquivo.IdFTPWork(ASender: TObject; AWorkMode:
TWorkMode; AWorkCount: Integer);
begin
  Progresso.Position := AWorkCount;
end;

end.

```

```

unit ufCadAvatar;

```

```

interface

```

```

uses

```

```

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

Dialogs, DB, ZAbstractRODataset, ZAbstractDataset, ZDataset, Grids, DBGrids, ExtCtrls, ToolWin, ComCtrls, ImgList, DBCtrls, StdCtrls, Mask, ExtDlgs;

type

```
TfCadAvatar = class(TForm)
  ToolBar: TToolBar;
  Panel: TPanel;
  Grid: TDBGrid;
  Query: TZQuery;
  tbInsert: TToolButton;
  tbDelete: TToolButton;
  tbFirst: TToolButton;
  tbPrior: TToolButton;
  tbNext: TToolButton;
  tbLast: TToolButton;
  ToolButton7: TToolButton;
  tbExit: TToolButton;
  lbTitulo: TLabel;
  edTitulo: TDBEdit;
  DataSource: TDataSource;
  ImageList: TImageList;
  StatusBar: TStatusBar;
  tbEdit: TToolButton;
  tbPost: TToolButton;
  ToolButton11: TToolButton;
  ToolButton1: TToolButton;
  ToolButton2: TToolButton;
  QueryChaveModelo: TIntegerField;
  QueryNomeArquivo: TStringField;
  QueryLinkDownload: TStringField;
```

```

edLink: TDBEdit;
lbLink: TLabel;
procedure tbExitClick(Sender: TObject);
procedure tbFirstClick(Sender: TObject);
procedure tbPriorClick(Sender: TObject);
procedure tbNextClick(Sender: TObject);
procedure tbLastClick(Sender: TObject);
procedure tbInsertClick(Sender: TObject);
procedure tbDeleteClick(Sender: TObject);
procedure tbEditClick(Sender: TObject);
procedure HabilitaCampos(b: boolean);
procedure tbPostClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure DataSourceDataChange(Sender: TObject; Field: TField);
procedure ToolButton1Click(Sender: TObject);
procedure QueryAfterScroll(DataSet: TDataSet);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  fCadAvatar: TfCadAvatar;

implementation

uses uDtModule;

{$R *.dfm}

```

```
procedure TfCadAvatar.tbExitClick(Sender: TObject);  
begin  
    modalresult := mrok;  
end;
```

```
procedure TfCadAvatar.tbFirstClick(Sender: TObject);  
begin  
    tbPost.Click;  
    Query.First;  
end;
```

```
procedure TfCadAvatar.tbPriorClick(Sender: TObject);  
begin  
    tbPost.Click;  
    Query.Prior;  
end;
```

```
procedure TfCadAvatar.tbNextClick(Sender: TObject);  
begin  
    tbPost.Click;  
    Query.Next;  
end;
```

```
procedure TfCadAvatar.tbLastClick(Sender: TObject);  
begin  
    tbPost.Click;  
    Query.Last;  
end;
```

```
procedure TfCadAvatar.tbInsertClick(Sender: TObject);  
var Qry: TZQuery;
```

```

    Chave: integer;
begin
    Query.Insert;
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select max(ChaveModelo) as m from Modelo');
    Qry.Open;
    if (Qry.FieldName('m').AsString='') then
        Chave := 1
    else
        Chave := Qry.fieldbyname('m').AsInteger+1;
    freeandnil(Qry);
    Query.FieldName('ChaveModelo').AsInteger := Chave;
    HabilitaCampos(true);
end;

```

```

procedure TfCadAvatar.tbDeleteClick(Sender: TObject);
begin
    if (Application.MessageBox('Tem certeza que deseja excluir este modelo?',
'Confirma exclusão', 4)=idYES) then
        begin
            Query.Delete;
            HabilitaCampos(false);
        end;
end;

```

```

procedure TfCadAvatar.tbEditClick(Sender: TObject);
begin
    Query.Edit;
    HabilitaCampos(true);
end;

```

```
procedure TfCadAvatar.HabilitaCampos(b: boolean);  
begin  
    edTitulo.Enabled := b;  
    edLink.Enabled := b;  
end;
```

```
procedure TfCadAvatar.tbPostClick(Sender: TObject);  
begin  
    HabilitaCampos(false);  
    if (Query.State=dsEdit) or (Query.State=dsInsert) then  
        Query.Post;  
end;
```

```
procedure TfCadAvatar.FormCreate(Sender: TObject);  
begin  
    Query.Active := true;  
    if (Query.RecordCount=0) then  
        begin  
            tbInsert.Click;  
        end  
    else  
        HabilitaCampos(false);  
end;
```

```
procedure TfCadAvatar.DataSourceDataChange(Sender: TObject; Field: TField);  
begin  
    StatusBar.Panels.Items[0].Text := 'Total de Avatares Cadastrados:  
' + inttostr(Query.RecordCount);  
end;
```



```

procedure TfCadAvatar.ToolButton1Click(Sender: TObject);
begin
    Query.Cancel;
    HabilitaCampos(false);
end;

```

```

procedure TfCadAvatar.QueryAfterScroll(DataSet: TDataSet);
begin
    HabilitaCampos(false);
end;

```

```

end.

```

```

unit ufCadAvatar;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ZAbstractRODataset, ZAbstractDataset, ZDataset, Grids,
    DBGrids, ExtCtrls, ToolWin, ComCtrls, ImgList, DBCtrls, StdCtrls, Mask,
    ExtDlgs;

```

```

type

```

```

    TfCadAvatar = class(TForm)
        ToolBar: TToolBar;
        Panel: TPanel;
        Grid: TDBGrid;
        Query: TZQuery;
        tblInsert: TToolButton;

```

```
tbDelete: TToolButton;  
tbFirst: TToolButton;  
tbPrior: TToolButton;  
tbNext: TToolButton;  
tbLast: TToolButton;  
ToolButton7: TToolButton;  
tbExit: TToolButton;  
lbTitulo: TLabel;  
edTitulo: TDBEdit;  
DataSource: TDataSource;  
ImageList: TImageList;  
StatusBar: TStatusBar;  
tbEdit: TToolButton;  
tbPost: TToolButton;  
ToolButton11: TToolButton;  
ToolButton1: TToolButton;  
ToolButton2: TToolButton;  
QueryChaveModelo: TIntegerField;  
QueryNomeArquivo: TStringField;  
QueryLinkDownload: TStringField;  
edLink: TDBEdit;  
lbLink: TLabel;  
procedure tbExitClick(Sender: TObject);  
procedure tbFirstClick(Sender: TObject);  
procedure tbPriorClick(Sender: TObject);  
procedure tbNextClick(Sender: TObject);  
procedure tbLastClick(Sender: TObject);  
procedure tbInsertClick(Sender: TObject);  
procedure tbDeleteClick(Sender: TObject);  
procedure tbEditClick(Sender: TObject);  
procedure HabilitaCampos(b: boolean);
```

```

    procedure tbPostClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure DataSourceDataChange(Sender: TObject; Field: TField);
    procedure ToolButton1Click(Sender: TObject);
    procedure QueryAfterScroll(DataSet: TDataSet);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    fCadAvatar: TfCadAvatar;

implementation

uses uDtModule;

{$R *.dfm}

procedure TfCadAvatar.tbExitClick(Sender: TObject);
begin
    modalresult := mrok;
end;

procedure TfCadAvatar.tbFirstClick(Sender: TObject);
begin
    tbPost.Click;
    Query.First;
end;

```

```

procedure TfCadAvatar.tbPriorClick(Sender: TObject);
begin
    tbPost.Click;
    Query.Prior;
end;

```

```

procedure TfCadAvatar.tbNextClick(Sender: TObject);
begin
    tbPost.Click;
    Query.Next;
end;

```

```

procedure TfCadAvatar.tbLastClick(Sender: TObject);
begin
    tbPost.Click;
    Query.Last;
end;

```

```

procedure TfCadAvatar.tbInsertClick(Sender: TObject);
var Qry: TZQuery;
    Chave: integer;
begin
    Query.Insert;
    Qry := TZQuery.Create(nil);
    Qry.Connection := DtModule.Connection;
    Qry.SQL.Add('Select max(ChaveModelo) as m from Modelo');
    Qry.Open;
    if (Qry.FieldName('m').AsString='') then
        Chave := 1
    else
        Chave := Qry.fieldbyname('m').AsInteger+1;

```

```

freeandnil(Qry);
Query.FieldName('ChaveModelo').AsInteger := Chave;
HabilitaCampos(true);
end;

```

```

procedure TfCadAvatar.tbDeleteClick(Sender: TObject);
begin
  if (Application.MessageBox('Tem certeza que deseja excluir este modelo?',
'Confirma exclusão', 4)=idYES) then
  begin
    Query.Delete;
    HabilitaCampos(false);
  end;
end;

```

```

procedure TfCadAvatar.tbEditClick(Sender: TObject);
begin
  Query.Edit;
  HabilitaCampos(true);
end;

```

```

procedure TfCadAvatar.HabilitaCampos(b: boolean);
begin
  edTitulo.Enabled := b;
  edLink.Enabled := b;
end;

```

```

procedure TfCadAvatar.tbPostClick(Sender: TObject);
begin
  HabilitaCampos(false);
  if (Query.State=dsEdit) or (Query.State=dsInsert) then

```

```
    Query.Post;  
end;
```

```
procedure TfCadAvatar.FormCreate(Sender: TObject);  
begin  
    Query.Active := true;  
    if (Query.RecordCount=0) then  
        begin  
            tblInsert.Click;  
        end  
    else  
        HabilitaCampos(false);  
end;
```

```
procedure TfCadAvatar.DataSourceDataChange(Sender: TObject; Field: TField);  
begin  
    StatusBar.Panels.Items[0].Text := 'Total de Avatares Cadastrados:  
' + inttostr(Query.RecordCount);  
end;
```

```
procedure TfCadAvatar.ToolButton1Click(Sender: TObject);  
begin  
    Query.Cancel;  
    HabilitaCampos(false);  
end;
```

```
procedure TfCadAvatar.QueryAfterScroll(DataSet: TDataSet);  
begin  
    HabilitaCampos(false);  
end;
```

end.

unit ufConfiguracoes;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, TFlatSpeedButtonUnit, ExtCtrls, ComCtrls, StdCtrls, uConfiguracoes,
Buttons, FileCtrl;

type

TfConfiguracoes = class(TForm)
 PageControl: TPageControl;
 tbGeral: TTabSheet;
 tbSons: TTabSheet;
 tbDiretorios: TTabSheet;
 pnMenu: TPanel;
 pnConfirmar: TPanel;
 btConfirmar: TFlatSpeedButton;
 pnCancelar: TPanel;
 btCancelar: TFlatSpeedButton;
 pnDados: TPanel;
 ckLogin: TCheckBox;
 ckMsg: TCheckBox;
 ckNudge: TCheckBox;
 ckConsole: TCheckBox;
 ckScreenshot: TCheckBox;
 tbControles: TTabSheet;
 ckSkin: TCheckBox;

ckSkinDialogo: TCheckBox;
GroupBox1: TGroupBox;
edFrente: TEdit;
Label1: TLabel;
Label2: TLabel;
edTras: TEdit;
Label3: TLabel;
edEsquerda: TEdit;
Label4: TLabel;
edDireita: TEdit;
Label5: TLabel;
cbModo: TComboBox;
GroupBox2: TGroupBox;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
edConversa: TEdit;
edAdicionar: TEdit;
edAnimacao: TEdit;
edScreenshot: TEdit;
Label10: TLabel;
edImages: TEdit;
Label11: TLabel;
edThumbs: TEdit;
Label12: TLabel;
edScreenshots: TEdit;
btImages: TBitBtn;
btThumbs: TBitBtn;
btScreenshots: TBitBtn;
lbLingua: TLabel;


```

cbLingua: TComboBox;
TabSheet1: TTabSheet;
ckLights: TCheckBox;
procedure FormCreate(Sender: TObject);
procedure btConfirmarClick(Sender: TObject);
procedure btCancelarClick(Sender: TObject);
procedure cbModoChange(Sender: TObject);
procedure btImagesClick(Sender: TObject);
procedure btThumbsClick(Sender: TObject);
procedure btScreenshotsClick(Sender: TObject);
private
  { Private declarations }
public
  procedure PreencheTela();
  procedure CarregaAbaGeral();
  procedure CarregaAbaSons();
  procedure CarregaAbaDiretorios();
  procedure CarregaAbaControles();
  procedure CarregaAbaMapa;
  procedure PreencheClasse();
  procedure GravaAbaGeral();
  procedure GravaAbaSons();
  procedure GravaAbaDiretorios();
  procedure GravaAbaControles();
  procedure GravaAbaMapa();
  { Public declarations }
end;

var
  fConfiguracoes: TfConfiguracoes;

```

implementation

{ \$R *.dfm }

{ TfConfiguracoes }

```
procedure TfConfiguracoes.CarregaAbaDiretorios;  
begin  
    edImages.Text := configuracoes.Dir_Images;  
    edThumbs.Text := configuracoes.Dir_Thumbs;  
    edScreenshots.Text := configuracoes.Dir_Screenshots;  
end;
```

```
procedure TfConfiguracoes.CarregaAbaGeral;  
begin  
    cbLingua.ItemIndex := configuracoes.Lingua;  
    ckSkin.Checked := configuracoes.Skin;  
    ckSkinDialogo.Checked := configuracoes.SkinDialogo;  
end;
```

```
procedure TfConfiguracoes.CarregaAbaSons;  
begin  
    ckLogin.Checked := configuracoes.SomLogin;  
    ckMsg.Checked := configuracoes.SomMsg;  
    ckNudge.Checked := configuracoes.SomNudge;  
    ckConsole.Checked := configuracoes.SomConsole;  
    ckScreenshot.Checked := configuracoes.SomScreenshot;  
end;
```

```
procedure TfConfiguracoes.GravaAbaDiretorios;  
begin
```

```

    configuracoes.Dir_Images := edImages.Text;
    configuracoes.Dir_Thumbs := edThumbs.Text;
    configuracoes.Dir_Screenshots := edScreenshots.Text;
end;

```

```

procedure TfConfiguracoes.GravaAbaGeral;
begin
    configuracoes.Lingua := cbLingua.ItemIndex;
    configuracoes.Skin := ckSkin.Checked;
    configuracoes.SkinDialogo := ckSkinDialogo.Checked;
end;

```

```

procedure TfConfiguracoes.GravaAbaSons;
begin
    configuracoes.SomLogin := ckLogin.Checked;
    configuracoes.SomMsg := ckMsg.Checked;
    configuracoes.SomNudge := ckNudge.Checked;
    configuracoes.SomConsole := ckConsole.Checked;
    configuracoes.SomScreenshot := ckScreenshot.Checked;
end;

```

```

procedure TfConfiguracoes.PreencheClasse;
begin
    GravaAbaGeral();
    GravaAbaSons();
    GravaAbaDiretorios();
    GravaAbaControles();
    GravaAbaMapa();
    configuracoes.Gravar;
end;

```

```
procedure TfConfiguracoes.PreencheTela;
```

```
begin
```

```
    CarregaAbaGeral();
```

```
    CarregaAbaSons();
```

```
    CarregaAbaDiretorios();
```

```
    CarregaAbaControles();
```

```
    CarregaAbaMapa();
```

```
end;
```

```
procedure TfConfiguracoes.FormCreate(Sender: TObject);
```

```
begin
```

```
    PreencheTela();
```

```
end;
```

```
procedure TfConfiguracoes.btConfirmarClick(Sender: TObject);
```

```
begin
```

```
    PreencheClasse();
```

```
    modalresult := mrok;
```

```
end;
```

```
procedure TfConfiguracoes.btCancelarClick(Sender: TObject);
```

```
begin
```

```
    modalresult := mrok;
```

```
end;
```

```
procedure TfConfiguracoes.CarregaAbaControles;
```

```
begin
```

```
    if (configuracoes.Tecla_Up=$00000011) then
```

```
    begin
```

```
        cbModo.ItemIndex := 0;
```

```
    end
```

```

else if (configuracoes.Tecla_Up=$000000C8) then
begin
    cbModo.ItemIndex := 1;
end;
cbModoChange(nil);
end;

```

```

procedure TfConfiguracoes.GravaAbaControles;
begin
    if (cbModo.ItemIndex=0) then
    begin
        configuracoes.Tecla_UP := $00000011;
        configuracoes.Tecla_Down := $0000001F;
        configuracoes.Tecla_Right := $00000020;
        configuracoes.Tecla_Left := $0000001E;
    end
    else if (cbModo.ItemIndex=1) then
    begin
        configuracoes.Tecla_UP := $000000C8;
        configuracoes.Tecla_Down := $000000D0;
        configuracoes.Tecla_Right := $000000CD;
        configuracoes.Tecla_Left := $000000CB;
    end;
end;

```

```

procedure TfConfiguracoes.cbModoChange(Sender: TObject);
begin
    if (cbModo.ItemIndex=0) then
    begin
        edFrente.Text := 'W';
        edTras.Text := 'S';
    end;
end;

```

```

    edEsquerda.Text := 'A';
    edDireita.Text := 'D';
end
else if (cbModo.ItemIndex=1) then
begin
    edFrente.Text := 'SETA PARA CIMA';
    edTras.Text := 'SETA PARA BAIXO';
    edEsquerda.Text := 'SETA PARA ESQUERDA';
    edDireita.Text := 'SETA PARA DIREITA';
end;
end;

procedure TfConfiguracoes.btImagesClick(Sender: TObject);
var S: string;
begin
    if (SelectDirectory('Selecionar uma pasta para armazenar as imagens dos
usuários do 3DBLAH', '\', S)) then
        edImages.Text := S;
end;

procedure TfConfiguracoes.btThumbsClick(Sender: TObject);
var S: string;
begin
    if (SelectDirectory('Selecionar uma pasta para armazenar as imagens dos mapas
do 3DBLAH', '\', S)) then
        edThumbs.Text := S;
end;

procedure TfConfiguracoes.btScreenshotsClick(Sender: TObject);
var S: string;
begin

```

```

    if (SelectDirectory('Selecionar uma pasta para armazenar as screenshots tiradas
no 3DBLAH', '\', S)) then

```

```

    edScreenshots.Text := S;
end;

```

```

procedure TfConfiguracoes.CarregaAbaMapa;

```

```

begin

```

```

    ckLights.Checked := configuracoes.Lights;
end;

```

```

procedure TfConfiguracoes.GravaAbaMapa;

```

```

begin

```

```

    configuracoes.Lights := ckLights.Checked;
end;

```

```

end.

```

```

unit ufConsole;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Buttons, MMSystem, uConfiguracoes;

```

```

type

```

```

    TfConsole = class(TForm)

```

```

        btOk: TBitBtn;

```

```

        btCancelar: TBitBtn;

```

```

        edConsole: TEdit;

```

```

    procedure btOkClick(Sender: TObject);
    procedure btCancelarClick(Sender: TObject);
private
    { Private declarations }
public
    procedure Console();
    { Public declarations }
end;

var
    fConsole: TfConsole;

implementation

uses ufPrincipal, ufServidores, TrueVision3D;

{$R *.dfm}

procedure TfConsole.btOkClick(Sender: TObject);
begin
    modalresult := mrok;
end;

procedure TfConsole.btCancelarClick(Sender: TObject);
begin
    modalresult := mrcancel;
end;

procedure TfConsole.Console;
var cheat: boolean;
begin

```



```
if (fPrincipal<>nil) then
  fPrincipal.PossuiFoco := false;
cheat := false;
if (fConsole.ShowModal=mrok) then
begin
  if (lowercase(edConsole.Text)='fps on') then
  begin
    if (fPrincipal<>nil) then
    begin
      fPrincipal.Globals.DisplayFPS(true);
      cheat := true;
    end;
  end
else if (lowercase(edConsole.Text)='fps off') then
begin
  if (fPrincipal<>nil) then
  begin
    fPrincipal.Globals.DisplayFPS(false);
    cheat := true;
  end;
end
else if (lowercase(edConsole.Text)='rendermode line') then
begin
  if (fPrincipal<>nil) then
  begin
    fPrincipal.Scene.SetRenderMode(TV_LINE);
    cheat := true;
  end;
end
else if (lowercase(edConsole.Text)='rendermode point') then
begin
```

```
if (fPrincipal<>nil) then
begin
  fPrincipal.Scene.SetRenderMode(TV_POINT);
  cheat := true;
end;
end
else if (lowercase(edConsole.Text)='rendermode normal') then
begin
  if (fPrincipal<>nil) then
  begin
    fPrincipal.Scene.SetRenderMode(TV_SOLID);
    cheat := true;
  end;
end
else if (lowercase(edConsole.Text)='showdata on') then
begin
  if (fPrincipal<>nil) then
  begin
    fPrincipal.MostraCoordenadas := true;
    cheat := true;
  end;
end
else if (lowercase(edConsole.Text)='showdata off') then
begin
  if (fPrincipal<>nil) then
  begin
    fPrincipal.MostraCoordenadas := false;
    cheat := true;
  end;
end;
end;
```

```

    if (cheat) and (configuracoes.SomConsole) then
        PlaySound(PChar(extractfilepath(application.ExeName)+'Sounds\cheat.wav'),
1, SND_ASYNC);
    end;
    if (fPrincipal<>nil) then
        fPrincipal.PossuiFoco := true;
end;

end.

```

```

unit ufDadosContato;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, StdCtrls, TFlatComboBoxUnit, Mask,
    TFlatMaskEditUnit, TFlatEditUnit, uUsuario;

```

```

type

```

```

    TfDadosContato = class(TForm)

```

```

        pnDados: TPanel;

```

```

        lbTelefone: TLabel;

```

```

        lbSexo: TLabel;

```

```

        lbPais: TLabel;

```

```

        lbNome: TLabel;

```

```

        lbNascimento: TLabel;

```

```

        lbEstadoCivil: TLabel;

```

```

        lbEstado: TLabel;

```

```

        lbCidade: TLabel;

```

```

lbCelular: TLabel;
edTelefone: TFlatEdit;
edNome: TFlatEdit;
edNascimento: TFlatMaskEdit;
edEstado: TFlatEdit;
edCidade: TFlatEdit;
edCelular: TFlatEdit;
cbSexo: TFlatComboBox;
cbPais: TFlatComboBox;
cbEstadoCivil: TFlatComboBox;
Image: TImage;
edEmail: TFlatEdit;
lbEmail: TLabel;
bordadir: TShape;
bordaesq: TShape;
bordainf: TShape;
bordasup: TShape;
edSexo: TFlatEdit;
edEstadoCivil: TFlatEdit;
edPais: TFlatEdit;
private
{ Private declarations }
public
procedure PreencheTela(Usuario: TUsuario);
{ Public declarations }
end;

var
fDadosContato: TfDadosContato;

implementation

```

```
{ $R *.dfm }
```

```
{ TfDadosContato }
```

```
procedure TfDadosContato.PreencheTela(Usuario: TUsuario);
```

```
begin
```

```
  try
```

```
    edNome.Text := Usuario.nome;
```

```
    edEmail.Text := Usuario.email;
```

```
    edNascimento.Text := datetostr(Usuario.nascimento);
```

```
    cbSexo.ItemIndex := Usuario.sexo;
```

```
    edSexo.Text := cbSexo.Text;
```

```
    cbEstadoCivil.ItemIndex := Usuario.estado_civil;
```

```
    edEstadoCivil.Text := cbEstadoCivil.Text;
```

```
    edCidade.Text := Usuario.cidade;
```

```
    edEstado.Text := Usuario.estado;
```

```
    cbPais.ItemIndex := Usuario.pais;
```

```
    edPais.Text := cbPais.Text;
```

```
    edTelefone.Text := Usuario.telefone;
```

```
    edCelular.Text := Usuario.celular;
```

```
    Image.Picture := Usuario.foto;
```

```
  except
```

```
  end;
```

```
end;
```

```
end.
```

```
unit ufDialogo;
```

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, ExtCtrls, Menus, StdCtrls, ComCtrls, ToolWin, ScktComp, uFuncoes,
 uConstantes, TFlatSpeedButtonUnit, WinSkinData, uUsuario, uConfiguracoes,
 IdBaseComponent, IdComponent, IdTCPConnection, IdTCPClient, IdFTP,
 MMSystem,
 ufrTransferencia, uFTP, ufrDownload;

type

TfDialogo = class(TForm)
 MainMenu: TMainMenu;
 Arquivo1: TMenuItem;
 Editar1: TMenuItem;
 Aes1: TMenuItem;
 Ajudas1: TMenuItem;
 pnMenu: TPanel;
 pnContato: TPanel;
 pnUsuario: TPanel;
 Splitter: TSplitter;
 reDialogo: TRichEdit;
 reUsuario: TRichEdit;
 pnBotoes: TPanel;
 StatusBar: TStatusBar;
 btEnviar: TButton;
 btApagar: TButton;
 pnDadosContato: TPanel;
 Fechar1: TMenuItem;
 Desfazer1: TMenuItem;
 N2: TMenuItem;

```
Recortar1: TMenuItem;  
Copiar1: TMenuItem;  
Colar1: TMenuItem;  
N3: TMenuItem;  
SelecionarTudo1: TMenuItem;  
picosdeAjuda1: TMenuItem;  
EnviarArquivo1: TMenuItem;  
btEnviarArquivo: TFlatSpeedButton;  
LimparConversa1: TMenuItem;  
btLimpar: TFlatSpeedButton;  
btSalvar: TFlatSpeedButton;  
Panel: TPanel;  
Timer: TTimer;  
btNudge: TFlatSpeedButton;  
btDados: TFlatSpeedButton;  
SalvarConversa2: TMenuItem;  
DadosdoContato1: TMenuItem;  
btAdd: TFlatSpeedButton;  
PopupMenu: TPopupMenu;  
Desfazer2: TMenuItem;  
Recortar2: TMenuItem;  
Copiar2: TMenuItem;  
Colar2: TMenuItem;  
N1: TMenuItem;  
N5: TMenuItem;  
SelecionarTudo2: TMenuItem;  
frTransferencia: TfrTransferencia;  
frDownload: TfrDownload;  
AdicionarContato1: TMenuItem;  
procedure btApagarClick(Sender: TObject);  
procedure ClientRead(Sender: TObject; Socket: TCustomWinSocket);
```

```

procedure FormCreate(Sender: TObject);
procedure btEnviarClick(Sender: TObject);
procedure Fechar1Click(Sender: TObject);
procedure Recortar1Click(Sender: TObject);
procedure Copiar1Click(Sender: TObject);
procedure Colar1Click(Sender: TObject);
procedure SelecionarTudo1Click(Sender: TObject);
procedure btLimparClick(Sender: TObject);
procedure TimerTimer(Sender: TObject);
procedure FormPaint(Sender: TObject);
procedure btNudgeClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure btDadosClick(Sender: TObject);
procedure btSalvarClick(Sender: TObject);
procedure Desfazer1Click(Sender: TObject);
procedure reUsuarioKeyPress(Sender: TObject; var Key: Char);
procedure btAddClick(Sender: TObject);
procedure btEnviarArquivoClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
private
  { Private declarations }
public
  Contato: TUsuario;
  ChaveUsuario: integer;
  ApelidoUsuario: string;
  TimerNudge: TTimer;
  SomMsg: boolean;
  SomNudge: boolean;
  procedure OnTimerNudge(Sender: TObject);
  procedure RecebeMsg(Msg: string);
  procedure RecebeNudge();

```



```

    procedure RecebeArquivo(Msg: string);
    procedure RecebeDadosContato(Msg: string);
    procedure AlteraNickUsuario(Msg: string);
    procedure AlteraNickContato(Msg: string);
    procedure DesabilitaAdd();
    procedure ForcaFoco();
    procedure TremeTela();
    procedure ContatoDesconectou();
    procedure ContatoReconectou();
    { Public declarations }
end;

var
    fDialogo: TfDialogo;

implementation

uses uDtModuleDialog, ufDadosContato;

{$R *.dfm}

procedure TfDialogo.btApagarClick(Sender: TObject);
begin
    reUsuario.Clear;
end;

procedure TfDialogo.ClientRead(Sender: TObject; Socket: TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Lista: TStringList;
    x: integer;

```

```
begin
  Lista := TStringList.Create;
  try
    DivideMsgs(Socket.ReceiveText, Lista);
    for x := 0 to Lista.Count-1 do
      begin
        DecodificaString(Lista[x],Codigo,Msg);
        if (Codigo=022) then
          AlteraNickUsuario(Msg)
        else if (Codigo=023) then
          AlteraNickContato(Msg)
        else if (Codigo=024) then
          RecebeDadosContato(Msg)
        else if (Codigo=028) then
          RecebeMsg(Msg)
        else if (Codigo=029) then
          ForcaFoco()
        else if (Codigo=039) then
          DesabilitaAdd()
        else if (Codigo=052) then
          RecebeNudge()
        else if (Codigo=61) then
          RecebeArquivo(Msg)
        else if (Codigo=64) then
          ContatoDesconectou()
        else if (Codigo=65) then
          ContatoReconectou();
      end;
    finally
      freeandnil(Lista);
    end;
```

end;

procedure TfDialogo.RecebeMsg(Msg: string);

var Lista: TStringList;

begin

if (SomMsg) then

PlaySound(PChar(extractfilepath(application.ExeName)+'Sounds\message.wav'),
1, SND_ASYNC);

Lista := TStringList.Create;

try

 SeparaString(Msg, Lista);

 reDialogo.SelStart := length(reDialogo.Text);

 reDialogo.Paragraph.FirstIndent := 2;

 reDialogo.SelAttributes.Color := clGray;

 reDialogo.Lines.Add(Lista[0]+' diz:');

 reDialogo.SelStart := length(reDialogo.Text);

 reDialogo.SelAttributes.Color := clBlack;

 reDialogo.Paragraph.FirstIndent := 12;

 reDialogo.Lines.Add(Lista[1]);

 reDialogo.SelStart := length(reDialogo.Text);

 reDialogo.Perform(EM_SCROLLCARET, 0, 0);

 Timer.Enabled := true;

finally

 freeandnil(Lista);

end;

end;

procedure TfDialogo.FormCreate(Sender: TObject);

begin

 Contato := TUsuario.Create;

```

if (ParamStr(2)='S') then
  SomMsg := true
else
  SomMsg := false;
if (ParamStr(3)='S') then
  SomNudge := true
else
  SomNudge := false;
if (ParamStr(4)='S') then
begin
  btAdd.Visible := false;
  AdicionarContato1.Visible := false;
end;
ChaveUsuario := strtoint(ParamStr(5));
ApelidoUsuario := ParamStr(6);
ApelidoUsuario := TrocaCaracter(ApelidoUsuario,SubstitutoEspaco,' ');
Contato.chave_usuario := strtoint(ParamStr(7));
Contato.apelido := ParamStr(8);
Contato.apelido := TrocaCaracter(Contato.Apelido,SubstitutoEspaco,' ');
DtModuleDialog.Client.OnRead := ClientRead;
DtModuleDialog.Client.Active := true;
Caption := Contato.apelido;
Application.Title := Contato.apelido;
end;

procedure TfDialogo.btEnviarClick(Sender: TObject);
var Texto: string;
begin
  if (length(RemoveCaracteres(reUsuario.Text, [Separador, SeparadorTamanho,
SubstitutoEspaco]))=0) then
    exit;

```

```

    Texto := ApelidoUsuario+separador+RemoveCaracteres(reUsuario.Text,
[Separador, SeparadorTamanho, SubstitutoEspaco]);

```

```

DtModuleDialog.Client.Socket.SendText(Protocolo('025'+inttostr(ChaveUsuario)+S
eparador+inttostr(Contato.chave_usuario)+Separador+Texto));

```

```

    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.Paragraph.FirstIndent := 2;
    reDialogo.SelAttributes.Color := clGray;
    reDialogo.Lines.Add(ApelidoUsuario+' diz:');
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.SelAttributes.Color := clBlack;
    reDialogo.Paragraph.FirstIndent := 12;
    reDialogo.Lines.Add(RemoveCaracteres(reUsuario.Text, [Separador,
SeparadorTamanho, SubstitutoEspaco]));
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.Perform(EM_SCROLLCARET, 0, 0);
    reUsuario.Clear;
    reUsuario.SetFocus;
end;

```

```

procedure TfDialogo.Fechar1Click(Sender: TObject);
begin
    application.Terminate;
end;

```

```

procedure TfDialogo.Recortar1Click(Sender: TObject);
begin
    reUsuario.CutToClipboard;
end;

```

```

procedure TfDialogo.Copiar1Click(Sender: TObject);

```

```
begin
    reUsuario.CopyToClipboard;
end;

procedure TfDialogo.Colar1Click(Sender: TObject);
begin
    reUsuario.PasteFromClipboard;
end;

procedure TfDialogo.SelecionarTudo1Click(Sender: TObject);
begin
    reUsuario.SelectAll;
end;

procedure TfDialogo.btLimparClick(Sender: TObject);
begin
    reDialogo.Clear;
end;

procedure TfDialogo.TimerTimer(Sender: TObject);
begin
    if (Active) then
    begin
        Timer.Enabled := false;
        exit;
    end;
    FlashWindow(Application.Handle, true);
end;

procedure TfDialogo.FormPaint(Sender: TObject);
begin
```

```

    Timer.Enabled := false;
end;

```

```

procedure TfDialogo.AlteraNickUsuario(Msg: string);
begin
    ApelidoUsuario := Msg;
end;

```

```

procedure TfDialogo.AlteraNickContato(Msg: string);
begin
    Contato.apelido := Msg;
    Caption := Contato.apelido;
    Application.Title := Contato.apelido;
end;

```

```

procedure TfDialogo.ForcaFoco;
begin
    ForceForegroundWindow(Application.Handle);
end;

```

```

procedure TfDialogo.btNudgeClick(Sender: TObject);
begin
    btNudge.Enabled := false;
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.Paragraph.FirstIndent := 2;
    reDialogo.SelAttributes.Color := clGray;
    reDialogo.Lines.Add('-----');
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.SelAttributes.Color := RGB(255,120,0);
    reDialogo.Lines.Add('Você acabou de pedir a atenção!');
    reDialogo.SelStart := length(reDialogo.Text);

```

```

reDialogo.SelAttributes.Color := clGray;
reDialogo.Lines.Add('-----');
reDialogo.SelStart := length(reDialogo.Text);
reDialogo.Perform(EM_SCROLLCARET, 0, 0);
TimerNudge := TTimer.Create(nil);
TimerNudge.OnTimer := OnTimerNudge;
TimerNudge.Interval := 8000;
TimerNudge.Enabled := true;

```

```

DtModuleDialog.Client.Socket.SendText(Protocolo('049'+inttostr(ChaveUsuario)+s
eparador+inttostr(Contato.chave_usuario)));
end;

```

```

procedure TfDialogo.OnTimerNudge(Sender: TObject);
begin
    btNudge.Enabled := true;
    freeandnil(Sender);
end;

```

```

procedure TfDialogo.RecebeNudge;
begin
    if (SomNudge) then
        PlaySound(PChar(extractfilepath(application.ExeName)+'Sounds\nudge.wav'),
1, SND_ASYNC);
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.Paragraph.FirstIndent := 2;
    reDialogo.SelAttributes.Color := clGray;
    reDialogo.Lines.Add('-----');
    reDialogo.SelStart := length(reDialogo.Text);
    reDialogo.SelAttributes.Color := RGB(255,120,0);
    reDialogo.Lines.Add(Contato.apelido+' acabou de pedir sua atenção!');

```



```

reDialogo.SelStart := length(reDialogo.Text);
reDialogo.SelAttributes.Color := clGray;
reDialogo.Lines.Add('-----');
ForcaFoco();
reDialogo.SelStart := length(reDialogo.Text);
reDialogo.Perform(EM_SCROLLCARET, 0, 0);
TremeTela();
end;

```

```

procedure TfDialogo.TremeTela;
var iLeft: integer;
    iTop: integer;
    x: integer;
begin
    iLeft := Left;
    iTop := Top;
    for x := 1 to 80 do
        begin
            if (x mod 2=0) then
                begin
                    Left := iLeft+Random(5);
                    Top := iTop+Random(5);
                end
            else
                begin
                    Left := iLeft-Random(5);
                    Top := iTop-Random(5);
                end;
            end;
        end;
    Left := iLeft;
    Top := iTop;

```

end;

```
procedure TfDialogo.RecebeDadosContato(Msg: string);
```

```
var Lista: TStringList;
```

```
begin
```

```
  Lista := TStringList.Create;
```

```
  try
```

```
    SeparaString(Msg, Lista);
```

```
    Contato.PreencheClasse(Lista);
```

```
  finally
```

```
    freeandnil(Lista);
```

```
  end;
```

```
  try
```

```
    Contato.CarregarImagem(nil);
```

```
    if (fDadosContato<>nil) then
```

```
      fDadosContato.PreencheTela(Contato);
```

```
  except
```

```
  end;
```

```
end;
```

```
procedure TfDialogo.FormDestroy(Sender: TObject);
```

```
begin
```

```
  freeandnil(Contato);
```

```
end;
```

```
procedure TfDialogo.btDadosClick(Sender: TObject);
```

```
begin
```

```
DtModuleDialog.Client.Socket.SendText(Protocolo('021'+inttostr(Contato.chave_usuario)));
```

```
fDadosContato := TfDadosContato.Create(nil);
```

```

fDadosContato.PreencheTela(Contato);
fDadosContato.ShowModal;
freeandnil(fDadosContato);
end;

procedure TfDialogo.btSalvarClick(Sender: TObject);
var Arquivo: string;
    Dia, Mes, Ano: Word;
    SaveDialog: TSaveDialog;
begin
    if (length(reDialogo.Text)>0) then
    begin
        DecodeDate(now, Ano, Mes, Dia);
        Arquivo := extractfilepath(application.ExeName)+'History\'+Contato.email+'
'+inttostr(Ano)+'-'+inttostr(Mes)+'-'+inttostr(Dia)+'.doc';
        SaveDialog := TSaveDialog.Create(nil);
        SaveDialog.Filter := 'Documento (.doc)|*.doc';
        SaveDialog.FileName := Arquivo;
        SaveDialog.InitialDir := extractfilepath(application.ExeName)+'History\';
        DtModuleDialog.SkinData.SkinFile:= '';
        DtModuleDialog.SkinData.Active := true;
        if (SaveDialog.Execute) then
            reDialogo.Lines.SaveToFile(Arquivo);
        DtModuleDialog.SkinData.SkinFile := DtModuleDialog.SkinData.SkinStore;
        DtModuleDialog.SkinData.Active := true;
        freeandnil(SaveDialog);
    end;
end;

procedure TfDialogo.Desfazer1Click(Sender: TObject);
begin

```

```

    reUsuario.Undo;
end;

```

```

procedure TfDialogo.reUsuarioKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key=#13) then
        begin
            Key := #0;
            btEnviar.Click;
        end;
    end;
end;

```

```

procedure TfDialogo.btAddClick(Sender: TObject);
begin

```

```

    DtModuleDialog.Client.Socket.SendText(Protocolo('004'+inttostr(Contato.chave_usuario)));
end;

```

```

procedure TfDialogo.DesabilitaAdd;
begin
    btAdd.Visible := false;
    AdicionarContato1.Visible := false;
    Caption := Contato.apelido;
    Application.Title := Contato.apelido;
end;

```

```

procedure TfDialogo.btEnviarArquivoClick(Sender: TObject);
var OpenDialog: TOpenDialog;
begin
    if (frTransferencia.Transferindo) then

```

```

    exit;
OpenDialog := TOpenDialog.Create(nil);
DtModuleDialog.SkinData.SkinFile:= '';
DtModuleDialog.SkinData.Active := true;
if (OpenDialog.Execute) then
begin
    frTransferencia.Visible := true;
    frTransferencia.NomeArquivo := OpenDialog.FileName;
    frTransferencia.IniciaTransferencia;
end;
DtModuleDialog.SkinData.SkinFile := DtModuleDialog.SkinData.SkinStore;
DtModuleDialog.SkinData.Active := true;
freeandnil(OpenDialog);
end;

procedure TfDialogo.RecebeArquivo(Msg: string);
var SaveDialog: TSaveDialog;
begin
    if (not frDownload.Transferindo) then
        if (Application.MessageBox(PAnsiChar(' Este usuário lhe enviou um arquivo.
'+#13+
                ' Arquivo: '+Msg+' '+#13+
                ' Deseja salvá-lo?'), 'Baixar Arquivo', 4)=idYES) then
begin
    DtModuleDialog.SkinData.SkinFile:= '';
    DtModuleDialog.SkinData.Active := true;
    SaveDialog := TSaveDialog.Create(nil);
    SaveDialog.FileName := Msg;
    SaveDialog.Filter := 'Arquivos do tipo
'+extractfileext(Msg)+'|*'+extractfileext(Msg);
    SaveDialog.InitialDir := extractfilepath(application.ExeName);

```

```

    if (SaveDialog.Execute) then
    begin
        frDownload.Visible := true;
        frDownload.NomeArquivoFTP := Msg;
        frDownload.NomeArquivo := SaveDialog.FileName;
        frDownload.IniciaTransferencia;
    end
    else
    begin
        DtModuleDialog.Client.Socket.SendText(Protocolo('062'+Msg));
    end;
    freeandnil(SaveDialog);
    DtModuleDialog.SkinData.SkinFile := DtModuleDialog.SkinData.SkinStore;
    DtModuleDialog.SkinData.Active := true;
end
else
begin
    DtModuleDialog.Client.Socket.SendText(Protocolo('062'+Msg));
end;
end;

procedure TfDialogo.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    if (frDownload.Transferindo) then
        frDownload.btCancelar.Click;
    if (frTransferencia.Transferindo) then
        frTransferencia.btCancelar.Click;
end;

procedure TfDialogo.ContatoDesconectou;
begin

```

```

reUsuario.Enabled := false;
btEnviar.Enabled := false;
btNudge.Enabled := false;
btEnviarArquivo.Enabled := false;
EnviarArquivo1.Enabled := false;
end;

```

```

procedure TfDialogo.ContatoReconnectou;
begin
    reUsuario.Enabled := true;
    btEnviar.Enabled := true;
    btNudge.Enabled := true;
    btEnviarArquivo.Enabled := true;
    EnviarArquivo1.Enabled := true;
end;

```

```

end.

```

```

unit ufLogin;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, OleCtrls, SHDocVw, ExtCtrls, uEditCorFoco, ufServidores,
    uFuncoes, ScktComp, uDtModuleClient, uConstantes, ShellApi, WinSkinData,
    uMensagemStatus, uConfiguracoes;

```

```

type

```

```

    TfLogin = class(TForm)

```

```

pnBrowser: TPanel;
pnLogin: TPanel;
WebBrowser: TWebBrowser;
lbUsuario: TLabel;
lbSenha: TLabel;
btEntrar: TButton;
btSair: TButton;
lbEsqueciSenha: TLabel;
btNovo: TButton;
lbDica01: TLabel;
lbDica02: TLabel;
edUsuario: TEditCorFoco;
edSenha: TEditCorFoco;
procedure FormCreate(Sender: TObject);
procedure btEntrarClick(Sender: TObject);
procedure btSairClick(Sender: TObject);
procedure ClientReadLogin(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientDisconnect(Sender: TObject; Socket: TCustomWinSocket);
procedure LoginAceito(Msg: string);
procedure LoginNegado(Msg: string);
procedure lbEsqueciSenhaClick(Sender: TObject);
procedure btNovoClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  fLogin: TfLogin;

```


implementation

```
{$R *.dfm}
```

```
// Se conecta ao servidor e abre o site de propagandas
```

```
procedure TfLogin.FormCreate(Sender: TObject);
```

```
begin
```

```
    DtModuleClient.Client.OnConnect := ClientConnect;
```

```
    DtModuleClient.Client.OnDisconnect := ClientDisconnect;
```

```
    DtModuleClient.Client.Open;
```

```
    DtModuleClient.Client.OnRead := ClientReadLogin;
```

```
WebBrowser.Navigate(extractfilepath(application.exename)+'atualizações\index.htm');
```

```
end;
```

```
// Verifica os campos e faz requisição de login para o servidor
```

```
procedure TfLogin.btEntrarClick(Sender: TObject);
```

```
var Msg: string;
```

```
begin
```

```
    if (length(edUsuario.Text)=0) then
```

```
    begin
```

```
        showmessage('Campo Usuário está vazio!');
```

```
        edUsuario.SetFocus;
```

```
    end
```

```
    else if (length(edSenha.Text)=0) then
```

```
    begin
```

```
        showmessage('Campo Senha está vazio!');
```

```
        edSenha.SetFocus;
```

```
    end
```

```

else
begin
    edUsuario.Enabled := false;
    edSenha.Enabled := false;
    btEntrar.Enabled := false;
    Msg := '001'+RemoveCaracteres(edUsuario.Text, [Separador,
SeparadorTamanho, SubstitutoEspaco])+separador+edSenha.Text;
    DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;
end;

```

```

procedure TfLogin.btSairClick(Sender: TObject);
begin
    application.Terminate;
end;

```

```

// Trata as mensagens recebidas do servidor
procedure TfLogin.ClientReadLogin(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Lista: TStringList;
    x: integer;
begin
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x], Codigo, Msg);
                if (Codigo=003) then

```

```

        LoginAceito(Msg)
    else if (Codigo=002) then
        LoginNegado(Msg);
    end;
finally
    freeandnil(Lista);
end;
end;

// Login Aceito, abre o form principal
procedure TfLogin.LoginAceito(Msg: string);
begin
    fServidores := TfServidores.Create(nil);
    try
        fServidores.PreencheClasseUsuario(Msg);
        fServidores.frListaContatos.lbNickUsuario.Caption :=
fServidores.Usuario.apelido;
        fServidores.frListaContatos.cbStatus.ItemIndex := fServidores.Usuario.status;
        fServidores.ShowModal;
    finally
        freeandnil(fServidores);
    end;
    try
        DtModuleClient.Client.OnRead := ClientReadLogin;
    finally
        edUsuario.Clear;
        edSenha.Clear;
        ClientConnect(nil, nil);
    end;
end;

```

```
// Login Negado, mostra msg com o motivo
procedure TfLogin.LoginNegado(Msg: string);
begin
    edUsuario.Enabled := true;
    edSenha.Enabled := true;
    btEntrar.Enabled := true;
    showmessage(' '+Msg+' ');
end;

procedure TfLogin.lbEsqueciSenhaClick(Sender: TObject);
begin
    ShellExecute(Application.Handle, nil,
PChar('http://www.3dblah.com.br/recuperasenha'), nil, nil, SW_SHOWNORMAL);
end;

procedure TfLogin.btNovoClick(Sender: TObject);
begin
    ShellExecute(Application.Handle, nil,
PChar('http://www.3dblah.com.br/cadastrar'), nil, nil, SW_SHOWNORMAL);
end;

procedure TfLogin.ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
    edUsuario.Enabled := true;
    edSenha.Enabled := true;
    btEntrar.Enabled := true;
    edUsuario.SetFocus;
end;

procedure TfLogin.ClientDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
```

```

begin
  if (fServidores<>nil) then
    begin
      showmessage('A conexão com o servidor foi perdida. A aplicação será
      fechada!');
      application.Terminate;
    end;
  end;

end.

```

```

unit ufmLogin;

```

```

interface

```

```

uses

```

```

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ScktComp, uConstantes, uFuncoes,
  ufmPrincipal;

```

```

type

```

```

  TfLogin = class(TForm)
    pnLogin: TPanel;
    lbUsuario: TLabel;
    lbSenha: TLabel;
    btEntrar: TButton;
    btSair: TButton;
    edUsuario: TEdit;
    edSenha: TEdit;
    procedure btSairClick(Sender: TObject);

```

```

procedure FormCreate(Sender: TObject);
procedure btEntrarClick(Sender: TObject);
procedure ClientReadLogin(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
procedure LoginAceito(Msg: string);
procedure LoginNegado(Msg: string);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  fLogin: TfLogin;

implementation

uses uDtModuleMap;

{$R *.dfm}

// -----
//  Ações dos Botões
// -----

procedure TfLogin.btEntrarClick(Sender: TObject);
var Msg: string;
begin
  if (length(edUsuario.Text)=0) then
  begin
    showmessage('Campo Usuário está vazio!');
  end;
end;

```

```

    edUsuario.SetFocus;
end
else if (length(edSenha.Text)=0) then
begin
    showmessage('Campo Senha está vazio!');
    edSenha.SetFocus;
end
else
begin
    edUsuario.Enabled := false;
    edSenha.Enabled := false;
    btEntrar.Enabled := false;
    Msg := '040'+edUsuario.Text+separador+edSenha.Text;
    DtModuleMap.Client.Socket.SendText(Protocolo(Msg));
end;
end;

procedure TfLogin.btSairClick(Sender: TObject);
begin
    application.Terminate;
end;

// -----
//  Fim das Ações dos Botões
// -----

// -----
//  Trata as mensagens recebidas do servidor principal
// -----

```

```

procedure TfLogin.ClientReadLogin(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Lista: TStringList;
    x: integer;
begin
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x], Codigo, Msg);
                if (Codigo=041) then
                    LoginAceito(Msg)
                else if (Codigo=042) then
                    LoginNegado(Msg);
            end;
        finally
            freeandnil(Lista);
        end;
    end;
end;

```

```

procedure TfLogin.LoginAceito(Msg: string);
begin
    fmPrincipal := TfmPrincipal.Create(nil);
    fmPrincipal.PreencheClasseMapa(Msg);
    fmPrincipal.ShowModal;
    DtModuleMap.Client.OnRead := ClientReadLogin;
    edUsuario.Clear;
    edSenha.Clear;

```



```

    ClientConnect(nil, nil);
    freeandnil(fmPrincipal);
end;

procedure TfLogin.LoginNegado(Msg: string);
begin
    edUsuario.Enabled := true;
    edSenha.Enabled := true;
    btEntrar.Enabled := true;
    showmessage(Msg);
end;

// -----
// Fim das mensagens recebidas do servidor principal
// -----

// -----
// Eventos dos componentes
// -----

procedure TfLogin.FormCreate(Sender: TObject);
begin
    DtModuleMap.Client.OnConnect := ClientConnect;
    DtModuleMap.Client.Open;
    DtModuleMap.Client.OnRead := ClientReadLogin;
end;

procedure TfLogin.ClientConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
    edUsuario.Enabled := true;
    edSenha.Enabled := true;

```

```

    btEntrar.Enabled := true;
    edUsuario.SetFocus;
end;

// -----
//  Fim dos eventos dos componentes
// -----

end.

```

```

unit ufmPrincipal;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ComCtrls, Grids, Menus, ExtCtrls, uMapa, uFuncoes, StdCtrls,
    ScktComp,
    TFlatSpeedButtonUnit, TFlatPanelUnit, TrueVision3D;

```

```

type

```

```

    TfmPrincipal = class(TForm)
        MainMenu: TMainMenu;
        Programa1: TMenuItem;
        Sair1: TMenuItem;
        RightPanel: TPanel;
        Grid: TStringGrid;
        StatusBar: TStatusBar;
        AtivarMapa1: TMenuItem;
        N1: TMenuItem;

```

```

Panel: TFlatPanel;
btKick: TFlatSpeedButton;
btKickAll: TFlatSpeedButton;
btSair: TFlatSpeedButton;
lbTitulo: TLabel;
edTitulo: TEdit;
lbDescricao: TLabel;
reDescricao: TRichEdit;
lbUsuarios: TLabel;
edUsuarios: TEdit;
lbHostname: TLabel;
lbArquivoMapa: TLabel;
lbArquivoTextura: TLabel;
edArquivoTextura: TEdit;
edArquivoMapa: TEdit;
edHostname: TEdit;
pnRight: TPanel;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure btSairClick(Sender: TObject);
procedure btKickClick(Sender: TObject);
procedure btKickAllClick(Sender: TObject);
private
    { Private declarations }
public
    Mapa: TMapa;
    TV: TTVEngine;
    Bsp: TTVBspTree;
    procedure PreencheClasseMapa(Msg: string);
    procedure AtualizaNumeroUsuarios();
    procedure SobreporClientRead(Sender: TObject; Socket: TCustomWinSocket);

```

```

    { Public declarations }
end;

var
    fmPrincipal: TfmPrincipal;

implementation

uses uDtModuleMap;

{$R *.dfm}

// -----
//  Ações dos Botões
// -----

procedure TfmPrincipal.btKickClick(Sender: TObject);
var p: pointer;
begin
    if (Grid.Row<>0) and
(DtModuleMap.Server.Socket.ActiveConnections>=Grid.Row) then
        begin
            p := DtModuleMap.Server.Socket.Connections[Grid.Row-1].Data;
            if (Application.MessageBox('    Tem certeza que deseja kickar este usuário?
', 'Kickar Usuário', 4)=idYES) then
                DtModuleMap.KickarUsuario(DtModuleMap.LocalizaCliente(p),");
        end;
end;

procedure TfmPrincipal.btKickAllClick(Sender: TObject);
var x: integer;

```

```

begin
  if (Application.MessageBox('    Tem certeza que deseja kickar todos os
usuários?    ', 'Kickar Usuários', 4)=idYES) then
    for x := 0 to DtModuleMap.Server.Socket.ActiveConnections-1 do
      DtModuleMap.KickarUsuario(x,"");
    end;

  procedure TfmPrincipal.btSairClick(Sender: TObject);
  begin
    application.Terminate;
  end;

  // -----
  //  Fim das Ações dos Botões
  // -----

  // -----
  //  Trata as mensagens recebidas do servidor principal
  // -----

  procedure TfmPrincipal.SobrepClientRead(Sender: TObject; Socket:
TCustomWinSocket);
  var Codigo: integer;
      Msg: string;
      Lista: TStringList;
      x: integer;
  begin
    Lista := TStringList.Create;
    try
      DivideMsgs(Socket.ReceiveText, Lista);
      for x := 0 to Lista.Count-1 do

```

```

begin
  DecodificaString(Lista[x],Codigo,Msg);
  if (Codigo=048) then
    DtModuleMap.SeDesconectaDoServidor(Msg);
  end;
finally
  freeandnil(Lista);
end;
end;

// -----
//  Fim das mensagens recebidas do servidor principal
// -----

// -----
//  Eventos dos componentes
// -----

procedure TfmPrincipal.FormCreate(Sender: TObject);
begin
  Mapa := TMapa.Create;
  Grid.Cells[0,0] := 'IP';
  Grid.Cells[1,0] := 'E-mail';
  Grid.Cells[2,0] := 'Conectado em';
  DtModuleMap.Client.OnRead := SobreporClientRead;
  DtModuleMap.Server.Active := true;
end;

procedure TfmPrincipal.FormDestroy(Sender: TObject);
begin
  freeandnil(Mapa);

```

```

    DtModuleMap.Server.Active := false;
end;

// -----
// Fim dos eventos dos componentes
// -----

// -----
// Outras funções
// -----

procedure TfmPrincipal.PreencheClasseMapa(Msg: string);
var Lista: TStringList;
begin
    Lista := TStringList.Create;
    try
        SeparaString(Msg, Lista);
        Mapa.PreencheClasse(Lista);
    finally
        freeandnil(Lista);
    end;
    edTitulo.Text := Mapa.Titulo;
    reDescricao.Text := Mapa.Descricao;
    AtualizaNumeroUsuarios();
    edHostname.Text := Mapa.HostName;
    edArquivoMapa.Text := Mapa.NomeArquivoMapa;
    edArquivoTextura.Text := Mapa.NomeArquivoTextura;

    TV := TTVEngine.Create(nil);
    Bsp := TTvBspTree.Create(nil);

```

```

Bsp.Load(extractfilepath(application.ExeName)+'\Maps'+Mapa.NomeArquivoMapa
, False, False, False, ", False, False);
DtModuleMap.PosInicio := Bsp.GetStartPlayerPosition;
freeandnil(Bsp);
freeandnil(TV);
end;

procedure TfmPrincipal.AtualizaNumeroUsuarios;
begin
    edUSuarios.Text := inttostr(Mapa.NumUsuarios)+' /
'+inttostr(Mapa.NumMaxUsuarios);
end;

// -----
//  Fim das outras funções
// -----

end.

unit ufPerfil;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, ComCtrls, uUsuario, ufrAvatar, ufrDadosPessoais,
    TFlatSpeedButtonUnit, ScktComp, uFuncoes, IdBaseComponent, IdComponent,
    IdTCPConnection, IdTCPClient, IdFTP, uFTP, uConfiguracoes, uConstantes;

```


type

```
TfPerfil = class(TForm)
  pnDados: TPanel;
  pnMenu: TPanel;
  StatusBar: TStatusBar;
  frDadosPessoais: TfrDadosPessoais;
  PageControl: TPageControl;
  tsDadosPessoais: TTabSheet;
  tsPersonagem: TTabSheet;
  frAvatar: TfrAvatar;
  btConfirmar: TFlatSpeedButton;
  btCancelar: TFlatSpeedButton;
  pnConfirmar: TPanel;
  pnCancelar: TPanel;
  IdFTP1: TIdFTP;
  procedure btConfirmarClick(Sender: TObject);
  procedure btCancelarClick(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
private
  { Private declarations }
public
  Usuario: TUsuario;
  procedure PreencheTelaDadosPessoais();
  procedure PreencheTelaAvatar();
  procedure PreencheClasse();
  procedure PreencheListaAvatares(Msg: string);
  procedure LiberaObjetosCriados();
  { Public declarations }
end;
```

var

```
fPerfil: TfPerfil;
```

```
implementation
```

```
uses uDtModuleClient;
```

```
{$R *.dfm}
```

```
procedure TfPerfil.PreencheTelaDadosPessoais();
```

```
begin
```

```
    frDadosPessoais.edNome.Text := Usuario.nome;
```

```
    frDadosPessoais.edNascimento.Text := datetostr(Usuario.nascimento);
```

```
    frDadosPessoais.cbSexo.ItemIndex := Usuario.sexo;
```

```
    frDadosPessoais.cbEstadoCivil.ItemIndex := Usuario.estado_civil;
```

```
    frDadosPessoais.edCidade.Text := Usuario.cidade;
```

```
    frDadosPessoais.edEstado.Text := Usuario.estado;
```

```
    frDadosPessoais.cbPais.ItemIndex := Usuario.pais;
```

```
    frDadosPessoais.edTelefone.Text := Usuario.telefone;
```

```
    frDadosPessoais.edCelular.Text := Usuario.celular;
```

```
    frDadosPessoais.Image.Picture := Usuario.foto;
```

```
    frDadosPessoais.ArquivolImagem := '';
```

```
end;
```

```
procedure TfPerfil.PreencheTelaAvatar();
```

```
begin
```

```
    frAvatar.edNick.Text := Usuario.apelido;
```

```
    frAvatar.cbStatus.ItemIndex := Usuario.status;
```

```
    frAvatar.lbNomeArquivo.Caption :=
```

```
extractfilepath(application.ExeName)+'Models\' + Usuario.modelo.nome_arquivo + '\  
Model.mdl';
```

```
    frAvatar.LoadAvatar(frAvatar.lbNomeArquivo.Caption);
```

```

    frAvatar.IndiceLista :=
frAvatar.ListaChavesAvatares.IndexOf(inttostr(Usuario.modelo.Chave_Modelo));
end;

procedure TfPerfil.PreencheClasse();
begin
    Usuario.nome := RemoveCaracteres(frDadosPessoais.edNome.Text, [Separador,
SeparadorTamanho, SubstitutoEspaco]);
    Usuario.nascimento := strtodate(frDadosPessoais.edNascimento.Text);
    Usuario.sexo := frDadosPessoais.cbSexo.ItemIndex;
    Usuario.estado_civil := frDadosPessoais.cbEstadoCivil.ItemIndex;
    Usuario.cidade := RemoveCaracteres(frDadosPessoais.edCidade.Text,
[Separador, SeparadorTamanho, SubstitutoEspaco]);
    Usuario.estado := RemoveCaracteres(frDadosPessoais.edEstado.Text,
[Separador, SeparadorTamanho, SubstitutoEspaco]);
    Usuario.pais := frDadosPessoais.cbPais.ItemIndex;
    Usuario.telefone := RemoveCaracteres(frDadosPessoais.edTelefone.Text,
[Separador, SeparadorTamanho, SubstitutoEspaco]);
    Usuario.celular := RemoveCaracteres(frDadosPessoais.edCelular.Text,
[Separador, SeparadorTamanho, SubstitutoEspaco]);
    Usuario.apelido := RemoveCaracteres(frAvatar.edNick.Text, [Separador,
SeparadorTamanho, SubstitutoEspaco]);
    Usuario.status := frAvatar.cbStatus.ItemIndex;
    Usuario.modelo.chave_modelo :=
strtoint(frAvatar.ListaChavesAvatares[frAvatar.IndiceLista]);
    Usuario.modelo.nome_arquivo :=
frAvatar.ListaNomesAvatares[frAvatar.IndiceLista];
end;

procedure TfPerfil.btConfirmarClick(Sender: TObject);
var IdFTP: TIdFTP;

```

```

Diretorios: TStringList;
x: integer;
begin
  if (FileExists(frDadosPessoais.ArquivolImagem)) then
  begin
    try
      IdFTP := TIdFTP.Create(nil);
      IdFTP.Username := FTP.FTPUser;
      IdFTP.Password := FTP.FTPPass;
      IdFTP.Host := FTP.FTPHost;
      IdFTP.Connect;
      Diretorios := TStringList.Create;
      GetDirList(FTP.FTPDir, Diretorios);
      for x := 0 to Diretorios.Count-1 do
        IdFTP.ChangeDir(Diretorios[x]);
      freeandnil(Diretorios);
      IdFTP.ChangeDir(FTP.FTPImgDir);
      IdFTP.Put(frDadosPessoais.ArquivolImagem,
FormatFloat('000000',Usuario.Chave_Usuario)+''.jpg');
      MoverJpeg(frDadosPessoais.ArquivolImagem,
configuracoes.Dir_Images+FormatFloat('000000',Usuario.chave_usuario)+''.jpg');

      Usuario.foto.LoadFromFile(configuracoes.Dir_Images+FormatFloat('000000',Usuar
io.chave_usuario)+''.jpg');
    finally
      IdFTP.Disconnect;
      freeandnil(IdFTP);
    end;
  end;
  modalresult := mrok;
end;

```

```

procedure TfPerfil.btCancelarClick(Sender: TObject);
begin
    modalresult := mrcancel;
end;

procedure TfPerfil.PreencheListaAvatares(Msg: string);
begin
    frAvatar.ListaNomesAvatares := TStringList.Create;
    frAvatar.ListaChavesAvatares := TStringList.Create;
    SeparaString2(Msg, frAvatar.ListaNomesAvatares,
frAvatar.ListaChavesAvatares);
    frAvatar.IndiceLista := -1;

    frAvatar.EngineCreate;
    frAvatar.EngineInit;
    frAvatar.Timer.Enabled := true;
    frAvatar.Image.Visible := true;

    frAvatar.MostraProximoAvatar();
end;

procedure TfPerfil.LiberaObjetosCriados;
begin
    freeandnil(frAvatar.ListaNomesAvatares);
    freeandnil(frAvatar.ListaChavesAvatares);
end;

procedure TfPerfil.FormDestroy(Sender: TObject);
begin
    LiberaObjetosCriados();

```

end;

end.

unit ufPrincipal;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, ExtCtrls, ufrListaContatos, TFlatSpeedButtonUnit, uFuncoes,
TrueVision3D, uMapa, Direct3D8, uUsuarioConectado, uConstantes, Math,
TVMath, MMSystem,
uConfiguracoes;

type

TfPrincipal = class(TForm)
 pn3D: TPanel;
 StatusBar: TStatusBar;
 pnListaContatos: TPanel;
 pnMenu: TPanel;
 btSair: TFlatSpeedButton;
 btAjuda: TFlatSpeedButton;
 pnBottom: TPanel;
 pnTop: TPanel;
 pnLeft: TPanel;
 pnRight: TPanel;
 btConversa: TFlatSpeedButton;
 btAddUsuario: TFlatSpeedButton;
 btScreenshot: TFlatSpeedButton;

```

Panel: TPanel;
procedure btSairClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure PerdeFoco(Sender: TObject);
procedure GanhaFoco(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormDeactivate(Sender: TObject);
procedure btConversaClick(Sender: TObject);
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure btAddUsuarioClick(Sender: TObject);
procedure btScreenshotClick(Sender: TObject);
private
  { Private declarations }
public
  Globals: TTvDelphiGlobals;
  TV: TTVEngine;
  Inp: TTvInputEngine;
  ScreenText: TTVScreen2dText;
  Scene: TTVScene;
  Bsp: TTVBspTree;
  Cam: TTVCamera;
  Mapa: TMapa;
  PlayerAngleY: Single;
  PlayerAngleYSpeed: Single;
  PlayerAngleYAim: Single;
  PlayerAngleX: Single;
  PlayerAngleXSpeed: Single;

```

```

PlayerAngleXAim: Single;
PlayerSpeed: Single;
PlayerPosition: D3DVECTOR;
PossuiFoco: boolean;
RotacionaCamera: boolean;
MostraCoordenadas: boolean;
Usuarios: array of TUsuarioConectado;
ChaveUsuarioMaisProximo: integer;
ApelidoUsuarioMaisProximo: string;
procedure EngineCreate();
procedure EngineEnd();
procedure EngineInit();
procedure Input();
procedure ConfereRaio();
procedure DestroiUsuarioMapa(ChaveUsuario: integer);
procedure ExibirCoordenadas;
procedure AppIdle(Sender: TObject; var Done: Boolean);
{ Public declarations }
end;

var
    fPrincipal: TfPrincipal;

implementation

uses uDtModuleClient, ufServidores;

{$R *.dfm}

// -----
//  Ações dos Botões

```



```
// -----
```

```
procedure TfPrincipal.btConversaClick(Sender: TObject);
```

```
begin
```

```
  if (not btConversa.Enabled) then
```

```
    exit;
```

```
  btConversa.Enabled := false;
```

```
  if (ChaveUsuarioMaisProximo>0) then
```

```
  begin
```

```
    with fServidores do
```

```
    begin
```

```
frListaContatos.IniciarConversa(Usuario.chave_usuario,Usuario.apelido,ChaveUsu  
arioMaisProximo,ApelidoUsuarioMaisProximo);
```

```
  end;
```

```
end;
```

```
  btConversa.Enabled := true;
```

```
end;
```

```
procedure TfPrincipal.btAddUsuarioClick(Sender: TObject);
```

```
begin
```

```
  if (not btAddUsuario.Enabled) then
```

```
    exit;
```

```
  btAddUsuario.Enabled := false;
```

```
  if (ChaveUsuarioMaisProximo>0) then
```

```
  begin
```

```
    if (Application.MessageBox(PChar('  Adicionar
```

```
  '+ApelidoUsuarioMaisProximo+' como amigo?  '), 'Adicionar Contato', 4)=idYES)
```

```
  then
```

```
DtModuleClient.Client.Socket.SendText(Protocolo('011'+inttostr(ChaveUsuarioMai
sProximo)));
```

```
end;
```

```
btAddUsuario.Enabled := true;
```

```
end;
```

```
procedure TfPrincipal.btScreenshotClick(Sender: TObject);
```

```
var Screenshot: string;
```

```
    Diretorio: string;
```

```
    x: integer;
```

```
begin
```

```
    if (not btScreenshot.Enabled) then
```

```
        exit;
```

```
    btScreenshot.Enabled := false;
```

```
    Diretorio := configuracoes.Dir_Screenshots;
```

```
    if (not directoryexists(Diretorio)) then
```

```
        ForceDirectories(Diretorio);
```

```
    x := 1;
```

```
    Screenshot := Diretorio+Mapa.Titulo+'_';
```

```
    while (fileexists(Screenshot+inttostr(x)+'.jpeg')) do
```

```
        inc(x);
```

```
    Screenshot := Screenshot+inttostr(x)+'.jpeg';
```

```
    if (configuracoes.SomScreenshot) then
```

```
        PlaySound(PChar(extractfilepath(application.ExeName)+'Sounds\click.wav'), 1,
```

```
SND_ASYNC);
```

```
    TV.ScreenShot(Screenshot);
```

```
    btScreenshot.Enabled := true;
```

```
end;
```

```

procedure TfPrincipal.btSairClick(Sender: TObject);
begin
    modalresult := mrok;
end;

// -----
//  Fim das Ações dos Botões
// -----

// -----
//  Eventos dos componentes
// -----

procedure TfPrincipal.FormCreate(Sender: TObject);
begin
    EngineCreate();
    Mapa := TMapa.Create;
    DtModuleClient.ClientMap.Socket.SendText(Protocolo('037'));
    GanhaFoco(nil);
    RotacionaCamera := false;
    MostraCoordenadas := false;
    ChaveUsuarioMaisProximo := 0;
    ApelidoUsuarioMaisProximo := '';
    Application.OnActivate := FormActivate;
    Application.OnDeactivate := FormDeactivate;
end;

procedure TfPrincipal.FormDestroy(Sender: TObject);
var x: integer;
begin

```

```

EngineEnd();
freeandnil(Mapa);
for x := 0 to length(Usuarios)-1 do
    freeandnil(Usuarios[x]);
end;

procedure TfPrincipal.FormActivate(Sender: TObject);
begin
    GanhaFoco(nil);
end;

procedure TfPrincipal.FormDeactivate(Sender: TObject);
begin
    PerdeFoco(nil);
end;

procedure TfPrincipal.FormMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    RotacionaCamera := true;
end;

procedure TfPrincipal.FormMouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    RotacionaCamera := false;
end;

// -----
//  Fim dos eventos dos componentes
// -----

```

```
// Cria os objetos da engine
procedure TfPrincipal.EngineCreate;
begin
    TV := TTVEngine.Create(nil);
    Scene := TTVScene.Create(nil);
    Bsp := TTvBspTree.Create(nil);
    Cam := TTvCamera.Create(nil);
    Globals := TTvDelphiGlobals.Create(nil);
    Inp := TTvInputEngine.Create(nil);
    ScreenText := TTVScreen2dText.Create(self);
end;
```

```
// Destroi os objetos da engine
procedure TfPrincipal.EngineEnd;
begin
    freeandnil(TV);
    freeandnil(Scene);
    freeandnil(Bsp);
    freeandnil(Cam);
    freeandnil(Globals);
    freeandnil(Inp);
    freeandnil(ScreenText);
end;
```

```
// Inicializa os objetos da engine
procedure TfPrincipal.EngineInit;
begin
    TV.Init3DWindowedMode(pn3D.handle,true);
    TV.SetAngleSystem(TV_ANGLE_DEGREE);
```

```

Bsp.OpenPackWAD(Mapa.NomeArquivoTextura, 0);
Bsp.Load(Mapa.NomeArquivoMapa, True, True, False, "", True, True);
Bsp.ClosePackWAD;
Bsp.SetParameters(configuracoes.Lights, True, TV_FRONT_CULL, False);

ScreenText.TextureFont_Create('default', 'verdana', 12, True, False, False,
False);

Cam.SetPosition(Bsp.GetStartPlayerPosition.x, Bsp.GetStartPlayerPosition.y,
Bsp.GetStartPlayerPosition.z);
end;

procedure TfPrincipal.Input;
var Posicao: D3DVector;
    mx, my: Integer;
    scroll: Integer;
    b1, b2, b3: SmallInt;
    ret: D3DVector;
    OldPlayerAngleYAim: Single;
begin
    if Inp.IsKeyPressed(TV_KEY_LEFTSHIFT) and (Inp.IsKeyPressed(TV_KEY_Z))
    then
        btScreenshot.Click;

        if Inp.IsKeyPressed(TV_KEY_LEFTSHIFT) and (Inp.IsKeyPressed(TV_KEY_C))
        then
            btConversa.Click;

            if Inp.IsKeyPressed(TV_KEY_LEFTSHIFT) and (Inp.IsKeyPressed(TV_KEY_A))
            then
                btAddUsuario.Click;

```

```
Posicao := Cam.GetPosition;
```

```
Inp.GetMouseState(mx, my, b1, b2, b3, scroll);
```

```
OldPlayerAngleYAim := PlayerAngleYAim;
```

```
if (RotacionaCamera) then
```

```
begin
```

```
    PlayerAngleYAim := PlayerAngleYAim + 0.003 * -mx;
```

```
    PlayerAngleXAim := PlayerAngleXAim + 0.003 * -my;
```

```
    if (PlayerAngleXAim>1.52) then
```

```
        PlayerAngleXAim := 1.52;
```

```
    if (PlayerAngleXAim<-1.52) then
```

```
        PlayerAngleXAim := -1.52;
```

```
end;
```

```
PlayerAngleXSpeed := (PlayerAngleXAim - PlayerAngleX) * 0.1;
```

```
PlayerAngleYSpeed := (PlayerAngleYAim - PlayerAngleY) * 0.1;
```

```
PlayerAngleX := PlayerAngleX + PlayerAngleXSpeed;
```

```
PlayerAngleY := PlayerAngleY + PlayerAngleYSpeed;
```

```
if Inp.IsKeyPressed(configuracoes.Tecla_Down) then
```

```
begin
```

```
    PlayerSpeed := PlayerSpeed - 0.15 * Tv.AccurateTimeElapsed;
```

```
end
```

```
else if Inp.IsKeyPressed(configuracoes.Tecla_Up) then
```

```
begin
```

```
    PlayerSpeed := PlayerSpeed + 0.15 * Tv.AccurateTimeElapsed;
```



```

+FloattoStr(PlayerAngleYAim)));

end;

procedure TfPrincipal.PerdeFoco(Sender: TObject);
begin
    PossuiFoco := false;
    fServidores.frListaContatos.cbStatus.Enabled := true;
end;

procedure TfPrincipal.GanhaFoco(Sender: TObject);
begin
    PossuiFoco := true;
    fServidores.frListaContatos.cbStatus.Enabled := false;
end;

procedure TfPrincipal.ConfereRaio;
var x: integer;
begin
    ChaveUsuarioMaisProximo := 0;
    ApelidoUsuarioMaisProximo := "";
    for x := 0 to length(Usuarios)-1 do
        begin
            if (Usuarios[x]<>nil) and
(Maths_GetDistance3D(Cam.getPosition.x,Cam.getPosition.y,Cam.getPosition.z,U
suarios[x].Personagem.getPosition.x,Usuarios[x].Personagem.getPosition.y,Usuari
os[x].Personagem.getPosition.z)<=100)
                and (Usuarios[x].Chave_Usuario<>fServidores.Usuario.chave_usuario) then
                begin
                    ScreenText.ACTION_BeginText;
                    ScreenText.TextureFont_DrawText(Usuarios[x].Apelido, pn3D.Left+15,
pn3D.Top+pn3D.Height-30, Globals.RGBA(1, 1, 1, 1), 'default');

```

```

    ScreenText.ACTION_EndText;
    ChaveUsuarioMaisProximo := Usuarios[x].Chave_Usuario;
    ApelidoUsuarioMaisProximo := Usuarios[x].Apelido;
    break;
end;
end;
end;

procedure TfPrincipal.DestroiUsuarioMapa(ChaveUsuario: integer);
var x: integer;
begin
    for x := 0 to length(Usuarios)-1 do
        begin
            if (Usuarios[x]<>nil) and (Usuarios[x].Chave_Usuario=ChaveUsuario) then
                begin
                    freeandnil(Usuarios[x]);
                    break;
                end
            end;
        end;
    end;
end;

procedure TfPrincipal.ExibirCoordenadas;
begin
    ScreenText.ACTION_BeginText;
    ScreenText.TextureFont_DrawText('x:
'+FormatFloat('###,###,###.0000',Cam.GetPosition.x), pn3D.Left+2,
pn3D.Top+10, Globals.RGBA(1, 1, 1, 1), 'default');
    ScreenText.TextureFont_DrawText('y:
'+FormatFloat('###,###,###.0000',Cam.GetPosition.y), pn3D.Left+2,
pn3D.Top+25, Globals.RGBA(1, 1, 1, 1), 'default');

```

```

    ScreenText.TextureFont_DrawText('z:
'+FormatFloat('###,###,###.0000',Cam.GetPosition.z), pn3D.Left+2,
pn3D.Top+40, Globals.RGBA(1, 1, 1, 1), 'default');
    ScreenText.ACTION_EndText;
end;

```

```

procedure TfPrincipal.AppIdle(Sender: TObject; var Done: Boolean);
var x: integer;
begin
    Done := false;
    TV.Clear(false);
    Bsp.Render;
    for x := 0 to length(Usuarios)-1 do
        if (Usuarios[x]<>nil) then
            Usuarios[x].Personagem.Render;
        if (PossuiFoco) then
            begin
                Input();
                ConfereRaio();
                if (MostraCoordenadas) then
                    ExibirCoordenadas();
            end;
        TV.RenderToScreen;
    end;

end.

```

```

unit ufrAvatar;

```

```

interface

```

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, TFlatComboBoxUnit, TFlatEditUnit,
TFlatSpeedButtonUnit, TrueVision3D, DBCtrls, Direct3d8, Menus;

type

```
TfrAvatar = class(TFrame)
  pnImagem: TPanel;
  pnDados: TPanel;
  pnModelo: TPanel;
  lbApelido: TLabel;
  lbStatus: TLabel;
  edNick: TFlatEdit;
  cbStatus: TFlatComboBox;
  btDireita: TFlatSpeedButton;
  btEsquerda: TFlatSpeedButton;
  Timer: TTimer;
  Image: TDBImage;
  lbNomeArquivo: TLabel;
  PopupMenu: TPopupMenu;
  Amarelo: TMenuItem;
  Azul: TMenuItem;
  Verde: TMenuItem;
  Vermelho: TMenuItem;
  Preto: TMenuItem;
  Branco: TMenuItem;
  procedure TimerTimer(Sender: TObject);
  procedure btEsquerdaClick(Sender: TObject);
  procedure btDireitaClick(Sender: TObject);
  procedure AmareloClick(Sender: TObject);
```

```

procedure AzulClick(Sender: TObject);
procedure BrancoClick(Sender: TObject);
procedure PretoClick(Sender: TObject);
procedure VerdeClick(Sender: TObject);
procedure VermelhoClick(Sender: TObject);
private
  { Private declarations }
public
  TV: TTVEngine;
  Avatar: TTVActor2;
  Scene: TTVScene;
  Angulo: double;
  ListaNomesAvatares: TStringList;
  ListaChavesAvatares: TStringList;
  IndiceLista: integer;
  Procedure EngineCreate;
  Procedure EngineInit;
  Procedure EngineEnd;
  Procedure LoadAvatar(NomeArquivo: string);
  Procedure MostraAvatarAnterior();
  Procedure MostraProximoAvatar();
  { Public declarations }
end;

```

implementation

uses ufServidores;

{ \$R *.dfm }

{ TfrAvatar }

```
// Cria os objetos da engine
procedure TfrAvatar.EngineCreate;
begin
    TV := TTVEngine.Create(self);
    Avatar := TTVActor2.Create(self);
    Scene := TTVScene.Create(self);

    Angulo := 0;
end;

// Destroi os objetos da engine
procedure TfrAvatar.EngineEnd;
begin
    freeandnil(TV);
    freeandnil(Avatar);
    freeandnil(Scene);
end;

// Inicializa os objetos da engine
procedure TfrAvatar.EngineInit;
begin
    TV.Init3DWindowedMode(Image.handle,true);
    TV.SetAngleSystem(TV_ANGLE_DEGREE);

    LoadAvatar(lbNomeArquivo.Caption);

    Scene.SetCamera(0, 0, -80, 0, 0, 0);
end;

// Carrega um modelo 3d
```

```

procedure TfrAvatar.LoadAvatar(NomeArquivo: string);
begin
  Avatar.Load(NomeArquivo,'Avatar',False,True);
  Avatar.SetPosition(0,-30,0);
  Avatar.PlayAnimation(12);
  Avatar.SetScale(1,1,1);
  lbNomeArquivo.Caption := NomeArquivo;
end;

```

// Rotaciona o modelo 3d e o renderiza

```

procedure TfrAvatar.TimerTimer(Sender: TObject);
begin
  TV.Clear(false);
  Avatar.SetRotation(0,Angulo,0);
  Avatar.Render;
  TV.RenderToScreen;
  Angulo := Angulo+0.5;
end;

```

// Procura o próximo modelo existente na máquina local

```

procedure TfrAvatar.MostraProximoAvatar;
var NomeArquivo: string;
begin
  repeat
    if (IndiceLista=ListaNomesAvatares.Count-1) then
      IndiceLista := 0
    else
      inc(IndiceLista);
      NomeArquivo :=
extractfilepath(application.ExeName)+'Models\' +ListaNomesAvatares.Strings[IndiceLista]+'Model.mdl';

```

```

    until fileexists(NomeArquivo);
    LoadAvatar(NomeArquivo);
end;

// Procura o modelo anterior existente na máquina local
procedure TfrAvatar.MostraAvatarAnterior;
var NomeArquivo: string;
begin
    repeat
        if (IndiceLista=0) then
            IndiceLista := ListaNomesAvatares.Count-1
        else
            dec(IndiceLista);
        NomeArquivo :=
extractfilepath(application.ExeName)+'Models\' +ListaNomesAvatares.Strings[Indic
eLista]+'Model.mdl';
        until fileexists(NomeArquivo);
        LoadAvatar(NomeArquivo);
    end;

procedure TfrAvatar.btEsquerdaClick(Sender: TObject);
begin
    MostraAvatarAnterior();
end;

procedure TfrAvatar.btDireitaClick(Sender: TObject);
begin
    MostraProximoAvatar();
end;

procedure TfrAvatar.AmareloClick(Sender: TObject);

```



```
begin
  Scene.SetSceneBackGround(1.0,1.0,0);
end;

procedure TfrAvatar.AzulClick(Sender: TObject);
begin
  Scene.SetSceneBackGround(0,0,1.0);
end;

procedure TfrAvatar.BrancoClick(Sender: TObject);
begin
  Scene.SetSceneBackGround(1.0,1.0,1.0);
end;

procedure TfrAvatar.PretoClick(Sender: TObject);
begin
  Scene.SetSceneBackGround(0,0,0);
end;

procedure TfrAvatar.VerdeClick(Sender: TObject);
begin
  Scene.SetSceneBackGround(0,1.0,0);
end;

procedure TfrAvatar.VermelhoClick(Sender: TObject);
begin
  Scene.SetSceneBackGround(1.0,0,0);
end;

end.
```

```
unit ufrDadosPessoais;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Mask, ExtCtrls, TFlatEditUnit, TFlatMaskEditUnit,  
TFlatComboBoxUnit, ExtDlgs, uFuncoes;
```

```
type
```

```
TfrDadosPessoais = class(TFrame)
```

```
  lbNome: TLabel;
```

```
  lbNascimento: TLabel;
```

```
  lbSexo: TLabel;
```

```
  lbEstadoCivil: TLabel;
```

```
  lbCidade: TLabel;
```

```
  lbEstado: TLabel;
```

```
  lbPais: TLabel;
```

```
  lbTelefone: TLabel;
```

```
  pnlImagem: TPanel;
```

```
  pnDados: TPanel;
```

```
  Image: TImage;
```

```
  cbSexo: TFlatComboBox;
```

```
  cbEstadoCivil: TFlatComboBox;
```

```
  cbPais: TFlatComboBox;
```

```
  edNascimento: TFlatMaskEdit;
```

```
  edNome: TFlatEdit;
```

```
  edCidade: TFlatEdit;
```

```
  edEstado: TFlatEdit;
```

```
  edTelefone: TFlatEdit;
```

```

bordaesq: TShape;
bordadir: TShape;
bordasup: TShape;
bordainf: TShape;
edCelular: TFlatEdit;
lbCelular: TLabel;
PictureDialog: TOpenPictureDialog;
procedure ImageClick(Sender: TObject);
private
  { Private declarations }
public
  ArquivolMagem: string;
  { Public declarations }
end;

implementation

uses uDtModuleClient;

{$R *.dfm}

procedure TfrDadosPessoais.ImageClick(Sender: TObject);
var Arq: string;
    Thumb: string;
begin
  DtModuleClient.SkinData.SkinFile:= '';
  DtModuleClient.SkinData.Active := true;
  if (PictureDialog.Execute) then
  begin
    if (lowercase(extractfileext(PictureDialog.FileName))='bmp') then
      BmpToJpg(PictureDialog.FileName, Arq)

```

```

else
  Arq := PictureDialog.FileName;
  Thumb := extractfilepath(application.ExeName)+'Temp1.jpg';
  CreateThumbnail(Arq, Thumb, 185, 185, clBlack);
  Image.Picture.LoadFromFile(Thumb);
  ArquivolImagem := Thumb;
  if (PictureDialog.FileName<>Arq) then
    DeleteFile(Arq);
  end;
  DtModuleClient.SkinData.SkinFile := DtModuleClient.SkinData.SkinStore;
  DtModuleClient.SkinData.Active := true;
end;

end.

unit ufrDownload;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, ExtCtrls, IdBaseComponent, IdComponent,
  IdTCPConnection, IdTCPClient, IdFTP, uFTP, uFuncoes, uConstantes,
  uBaixaArquivo;

type
  TfrDownload = class(TFrame)
    ProgressBar: TProgressBar;
    Lb: TLabel;
    btConcluido: TBitBtn;
  end;

```

```

    btCancelar: TBitBtn;
    pnRight: TPanel;
    Panel: TPanel;
    procedure IdFTPWorkEnd(Sender: TObject; AWorkMode: TWorkMode);
    procedure btConcluidoClick(Sender: TObject);
    procedure btCancelarClick(Sender: TObject);
private
    { Private declarations }
public
    Transferindo: boolean;
    NomeArquivo: string;
    NomeArquivoFTP: string;
    Th: TBaixaArquivo;
    procedure IniciaTransferencia();
    { Public declarations }
end;

implementation

uses uDtModuleDialog, ufDialogo;

{$R *.dfm}

{ TfrDownload }

procedure TfrDownload.IdFTPWorkEnd(Sender: TObject; AWorkMode:
TWorkMode);
begin
    try
        Transferindo := false;
        ProgressBar.Position := ProgressBar.Max;

```

```

    Lb.Caption := 'Concluído';
    btCancelar.Visible := false;
    btConcluido.Visible := true;
    Refresh;
    DtModuleDialog.Client.Socket.SendText(Protocolo('062'+NomeArquivoFTP));
    NomeArquivo := '';
except
end;
end;

```

```

procedure TfrDownload.IniciaTransferencia;
begin
    Transferindo := true;
    lb.Caption := 'Baixando '+extractfilename(NomeArquivo);
    visible := true;
    Th := TBaixaArquivo.Criar(NomeArquivoFTP, NomeArquivo, ProgressBar);
    Th.IdFTP.OnWorkEnd := IdFTPWorkEnd;
    Th.Resume;
end;

```

```

procedure TfrDownload.btConcluidoClick(Sender: TObject);
begin
    visible := false;
end;

```

```

procedure TfrDownload.btCancelarClick(Sender: TObject);
begin
    try
        Transferindo := false;
        Th.IdFTP.OnWorkEnd := nil;
        Th.IdFTP.Abort;
    end;
end;

```

```

finally
    btConcluido.Click;
    DtModuleDialog.Client.Socket.SendText(Protocolo('062'+NomeArquivoFTP));
end;
end;

end.

```

```

unit ufrListaContatos;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, TFlatComboBoxUnit,
    uTolderlundStringGrid,
    TFlatPanelUnit, jpeg, TFlatSpeedButtonUnit, uUsuario, ImgList,
    uAdicionarContato, uFuncoes, uConfiguracoes, uConstantes;

```

```

type

```

```

    TfrListaContatos = class(TFrame)
        Grid: TStringGrid;
        imgUsuario: TImage;
        lbNick: TLabel;
        cbStatus: TFlatComboBox;
        lbStatus: TLabel;
        lbNickUsuario: TLabel;
        pnBotoes: TFlatPanel;
        Shape1: TShape;
        Shape2: TShape;
    end;

```

```

Shape3: TShape;
Shape4: TShape;
btAdd: TFlatSpeedButton;
btRemove: TFlatSpeedButton;
ImageList: TImageList;
Panel: TFlatPanel;
procedure GridDrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
procedure btRemoveClick(Sender: TObject);
procedure btAddClick(Sender: TObject);
procedure GridMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure GridContextPopup(Sender: TObject; MousePos: TPoint;
  var Handled: Boolean);
private
  { Private declarations }
public
  Contatos: array of TUsuario;
  ContatoCount: integer;
  Row: integer;
  procedure NovoContato();
  procedure DeletarContato(Posicao: integer);
  function IndexOf(ChaveContato:integer): integer;
  function IndexOfPos(Posicao: integer): integer;
  procedure ReOrganizaPosicoes(PosAnt: integer; Pos: integer);
  procedure IniciarConversa(ChaveUsuario: integer; ApelidoUsuario: string;
ChaveContato: integer; ApelidoContato: string);
  { Public declarations }
end;

implementation

```



```
uses uDtModuleClient;
```

```
{ $R *.dfm }
```

```
{ TfrListaContatos }
```

```
procedure TfrListaContatos.DeletarContato(Posicao: integer);
```

```
var x: integer;
```

```
    Ind: integer;
```

```
    Pos: integer;
```

```
begin
```

```
    Ind := IndexOfPos(Posicao);
```

```
dtModuleClient.Client.Socket.SendText(Protocolo('013'+inttostr(Contatos[Ind].chav  
e_usuario)));
```

```
    Pos := Contatos[Ind].posicao;
```

```
    for x := Ind to ContatoCount-2 do
```

```
    begin
```

```
        Contatos[x] := Contatos[x+1];
```

```
    end;
```

```
    dec(ContatoCount);
```

```
    setlength(Contatos, ContatoCount);
```

```
    for x := 0 to ContatoCount-1 do
```

```
        if (Contatos[x].posicao>Pos) then
```

```
            Contatos[x].posicao := Contatos[x].posicao-1;
```

```
        TStringGridHack(Grid).DeleteRow(Posicao-1);
```

```
    end;
```

```
procedure TfrListaContatos.NovoContato;
```

```
begin
```

```

inc(ContatoCount);
setlength(Contatos, ContatoCount);
Contatos[ContatoCount-1] := TUsuario.Create;
end;

```

```

procedure TfrListaContatos.GridDrawCell(Sender: TObject; ACol,
  ARow: Integer; Rect: TRect; State: TGridDrawState);
var Texto: string;
    AlturaTexto, Y: integer;
    Indice: integer;
begin
  if (ContatoCount>ARow) then
  begin
    Indice := IndexOfPos(ARow+1);
    if (ARow=Grid.Row) then
    begin
      Grid.Canvas.Brush.Color := clSkyBlue;
      Grid.Canvas.Font.Style := Grid.Canvas.Font.Style + [fsBold];
      Grid.Canvas.FillRect(Rect);
    end;
    if (ACol=0) then
    begin
      ImageList.Draw(Grid.Canvas,Rect.Left+5,Rect.Top+7,Contatos[Indice].sexo);
    end
    else if (ACol=1) then
    begin
      ImageList.Draw(Grid.Canvas,Rect.Left+5,Rect.Top+7,Contatos[Indice].pais+2);
    end
    else
    begin
      if (Acol=3) then

```

```

begin
  if (Contatos[Indice].status=4) then
    Grid.Canvas.Font.Color := clGreen
  else if (Contatos[Indice].status=-1) then
    Grid.Canvas.Font.Color := clMaroon
  else
    Grid.Canvas.Font.Color := clOlive;
  end;
  Texto := Grid.Cells[ACol, ARow];
  AlturaTexto := Grid.Canvas.TextHeight(Texto);
  Y := (Rect.Top + (Rect.Bottom - Rect.Top) div 2 - AlturaTexto div 2)-1;
  Grid.Canvas.TextRect(Rect, Rect.Left, Y, Texto);
end;
end;
end;

procedure TfrListaContatos.btRemoveClick(Sender: TObject);
begin
  if (ContatoCount>0) and (Application.MessageBox(' Tem certeza que deseja
remover este contato? ', 'Confirma remoção', 4)=idYES) then
    DeletarContato(Grid.Row+1);
end;

procedure TfrListaContatos.btAddClick(Sender: TObject);
var form: TfAdicionarContato;
    email: string;
begin
  form := TfAdicionarContato.Create(nil);
  if form.AbrirForm(email) then
    begin
      DtModuleClient.Client.Socket.SendText(Protocolo('012'+email));
    end;
  end;
end;

```

```

end;
freeandnil(form);
end;

```

```

function TfrListaContatos.IndexOf(ChaveContato: integer): integer;
var x: integer;
begin
    result := -1;
    for x := 0 to ContatoCount-1 do
        if (Contatos[x].chave_usuario=ChaveContato) then
            begin
                result := x;
                break;
            end;
    end;
end;

```

```

procedure TfrListaContatos.IniciarConversa(ChaveUsuario: integer;
ApelidoUsuario: string; ChaveContato: integer; ApelidoContato: string);
var Index: integer;
    Skin: string;
    Amigo: string;
    SomNudge: string;
    SomMsg: string;
begin
    Index := DtModuleClient.ChavesUsuarios.IndexOf(inttostr(ChaveContato));
    if (Index=-1) then
        begin
            DtModuleClient.ChavesUsuarios.Add(inttostr(ChaveContato));
            DtModuleClient.EstadosUsuarios.Add('EM ESPERA');
            if (IndexOf(ChaveContato)=-1) then
                Amigo := 'N'
            end;
        end;
    end;
end;

```

```

else
  Amigo := 'S';
if (configuracoes.SomMsg) then
  SomMsg := 'S'
else
  SomMsg := 'N';
if (configuracoes.SomNudge) then
  SomNudge := 'S'
else
  SomNudge := 'N';
if (configuracoes.SkinDialogo) then
  Skin := 'S'
else
  Skin := 'N';
  ExecuteProgram(extractfilepath(application.ExeName)+'Conversa.exe',Skin+'
'+SomMsg+' '+SomNudge+' '+Amigo+' '+inttostr(ChaveUsuario)+'
'+TrocaCaracter(ApelidoUsuario,' ',SubstitutoEspaco)+' '+inttostr(ChaveContato)+'
'+TrocaCaracter(ApelidoContato,' ',SubstitutoEspaco));
end
else
begin
  if (DtModuleClient.EstadosUsuarios[Index]<>'EM ESPERA') then
    DtModuleClient.Server.Socket.Connections[Index].SendText(Protocolo('029'));
  end;
end;

procedure TfrListaContatos.ReOrganizaPosicoes(PosAnt: integer; Pos: integer);
var x: integer;
begin
  for x := 0 to ContatoCount-1 do
    if (Contatos[x].posicao=PosAnt) then

```

```

    Contatos[x].posicao := Pos
  else if (PosAnt>Pos) and (Contatos[x].posicao>=Pos) and
    (Contatos[x].posicao<=PosAnt) then
    Contatos[x].posicao := Contatos[x].posicao+1
  else if (PosAnt<Pos) and (Contatos[x].posicao<=Pos) and
    (Contatos[x].posicao>=PosAnt) then
    Contatos[x].posicao := Contatos[x].posicao-1
end;

```

```

function TfrListaContatos.IndexOfPos(Posicao: integer): integer;
var x: integer;
begin
  result := -1;
  for x := 0 to ContatoCount-1 do
    if (Contatos[x].posicao=Posicao) then
      begin
        result := x;
        break;
      end;
  end;
end;

```

```

procedure TfrListaContatos.GridMouseMove(Sender: TObject; Shift: TShiftState;
X, Y: Integer);
var Linha: integer;
    Indice: integer;
begin
  Grid.MouseToCell(X, Y, Indice, Linha);
  if (Row <> Linha) then
    begin
      Row := Linha;
      Application.CancelHint;
    end;
end;

```

```

    Indice := IndexOfPos(Row+1);
    if (Indice>-1) then
        Grid.Hint := Contatos[Indice].apelido+#13+'('+Contatos[Indice].email+')';
    end;
end;

procedure TfrListaContatos.GridContextPopup(Sender: TObject; MousePos:
TPoint; var Handled: Boolean);
var Coluna: integer;
    Linha: integer;
begin
    Grid.MouseToCell(MousePos.X, MousePos.Y, Coluna, Linha);
    if (Linha>=0) then
        Grid.Row := Linha;
    end;

end.

unit ufrMapa;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, StdCtrls, ComCtrls, jpeg, uMapa, IdBaseComponent,
    uFuncoes,
    IdComponent, IdTCPConnection, IdTCPClient, IdFTP, uFTP, uConfiguracoes,
    uBaixaImagem;

type

```

```

TfrMapa = class(TFrame)
  Image: TImage;
  lbTitulo: TLabel;
  lbQtd: TLabel;
  reDescricao: TRichEdit;
  Panel: TPanel;
private
  { Private declarations }
public
  Mapa: TMapa;
  Selecionado: boolean;
  Th: TBaixaImgMapa;
  procedure SeleccionaFrame;
  procedure DeSeleccionaFrame;
  procedure CarregarImagem;
  constructor Create(Owner: TComponent); override;
  destructor Destroy; override;
  { Public declarations }
end;

implementation

{$R *.dfm}

{ TfrMapa }

procedure TfrMapa.SeleccionaFrame;
begin
  Panel.color := clSkyBlue;
  reDescricao.Color := clSkyBlue;
  Selecionado := true;

```


end;

procedure TfrMapa.DeSelecionaFrame;

begin

 Panel.color := clBtnFace;

 reDescricao.Color := clBtnFace;

 Selecionado := false;

end;

procedure TfrMapa.CarregarImagem;

begin

 Th := TBaixaImagem.Criar(Image);

 Th.ChaveServidor := Mapa.ChaveServidor;

 Th.Resume;

end;

constructor TfrMapa.Create(Owner: TComponent);

begin

 inherited;

 Mapa := TMapa.Create;

end;

destructor TfrMapa.Destroy;

begin

 if (Th<>nil) then

 Th.Terminate;

 freeandnil(Mapa);

 inherited;

end;

end.

```
unit ufrTransferencia;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, Buttons, ComCtrls, ExtCtrls, IdBaseComponent, IdComponent,  
  IdTCPConnection, IdTCPClient, IdFTP, uFTP, uFuncoes, uConstantes,  
  uEnviaArquivo,  
  IdExplicitTLSClientServerBase;
```

```
type
```

```
  TfrTransferencia = class(TFrame)
```

```
    ProgressBar: TProgressBar;
```

```
    Lb: TLabel;
```

```
    btConcluido: TBitBtn;
```

```
    btCancelar: TBitBtn;
```

```
    pnRight: TPanel;
```

```
    Panel: TPanel;
```

```
    IdFTP1: TIdFTP;
```

```
    procedure IdFTPWorkEnd(ASender: TObject; AWorkMode: TWorkMode);
```

```
    procedure btConcluidoClick(Sender: TObject);
```

```
    procedure btCancelarClick(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    Transferindo: boolean;
```

```
    NomeArquivo: string;
```

```
    NomeArquivoFTP: string;
```

```

Cancelou: boolean;
Th: TEnviaArquivo;
procedure IniciaTransferencia();
{ Public declarations }
end;

```

implementation

```

uses uDtModuleDialog, ufDialogo;

```

```

{$R *.dfm}

```

```

{ TfrTransferencia }

```

```

procedure TfrTransferencia.IdFTPWorkEnd(ASender: TObject; AWorkMode:
TWorkMode);
begin
  try
    Transferindo := false;
    NomeArquivo := '';
    ProgressBar.Position := ProgressBar.Max;
    Lb.Caption := 'Concluído';
    btCancelar.Visible := false;
    btConcluido.Visible := true;
    Refresh;
    if (not Cancelou) then
      DtModuleDialog.Client.Socket.SendText(Protocolo('055'+inttostr((Owner as
TfDialogo).ChaveUsuario)+separador+inttostr((Owner as
TfDialogo).Contato.chave_usuario)+separador+th.Arquivo));
  except
  end;

```

end;

```

procedure TfrTransferencia.IniciaTransferencia;
begin
  if (FileExists(NomeArquivo)) then
  begin
    Cancelou := false;
    Transferindo := true;
    lb.Caption := 'Enviando '+extractfilename(NomeArquivo);
    visible := true;
    Th := TEnviaArquivo.Criar(NomeArquivo, ProgressBar);
    Th.IdFTP.OnWorkEnd := IdFTPWorkEnd;
    Th.Resume;
  end;
end;

```

```

procedure TfrTransferencia.btConcluidoClick(Sender: TObject);
begin
  visible := false;
end;

```

```

procedure TfrTransferencia.btCancelarClick(Sender: TObject);
begin
  try
    Cancelou := true;
    Transferindo := false;
    Th.IdFTP.OnWorkEnd := nil;
    Th.IdFTP.Abort;
  finally
    btConcluido.Click;
  end;
end;

```

end;

end.

unit ufServidores;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, ComCtrls, ExtCtrls, ufrListaContatos, TFlatSpeedButtonUnit, uPing,
 ufrMapa, ufPerfil, uUsuario, uFuncoes, ScktComp, TFlatPanelUnit, ufConsole,
 Math,
 uConstantes, TrueVision3D, StdCtrls, uMapa, ufPrincipal, ShellApi,
 uUsuarioConectado,
 uTolderlundStringGrid, MSNPopUp, IdBaseComponent, IdComponent,
 MMSystem,
 IdTCPConnection, IdTCPClient, IdFTP, uConfiguracoes, ufConfiguracoes, uFTP;

type

TfServidores = class(TForm)
 pnMapas: TPanel;
 StatusBar: TStatusBar;
 pnListaContatos: TPanel;
 pnMenu: TPanel;
 frListaContatos: TfrListaContatos;
 btPerfil: TFlatSpeedButton;
 btConfig: TFlatSpeedButton;
 btConectar: TFlatSpeedButton;
 btAjuda: TFlatSpeedButton;

```

btLogoff: TFlatSpeedButton;
btAtualizar: TFlatSpeedButton;
btSair: TFlatSpeedButton;
btEsquerda: TFlatSpeedButton;
btDireita: TFlatSpeedButton;
procedure btPerfilClick(Sender: TObject);
procedure btLogoffClick(Sender: TObject);
procedure ClientReadLogin(Sender: TObject; Socket: TCustomWinSocket);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure btSairClick(Sender: TObject);
procedure frListaContatoscbStatusChange(Sender: TObject);
procedure frListaContatosGridDbClick(Sender: TObject);
procedure btDireitaClick(Sender: TObject);
procedure btEsquerdaClick(Sender: TObject);
procedure btAtualizarClick(Sender: TObject);
procedure btConectarClick(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
procedure MSNPopUpClick(Sender: TObject);
procedure btConfigClick(Sender: TObject);
procedure frListaContatosGridKeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
private
  { Private declarations }
public
  Usuario: TUsuario;
  PaginaServidores: integer;
  TotalPaginasServidores: integer;
  //VariavelPraEvitarBug: integer;
  Servidor: array[0..7] of TfrMapa;
  Incrementavel: integer;

```

```
FrameSelecionado: integer;
MSNPoUp: TMSNPoUp;
procedure PreencheClasseUsuario(Msg: string);
procedure SolicitaListaContatos();
procedure SolicitaListaPedidos();
procedure SolicitaListaArquivos();
procedure SolicitaPaginaServidores(Pagina: integer);
procedure SolicitaDadosGerais();
procedure PreencheContatosNomes(Msg: string);
procedure AmigoAdicionado(Msg: string);
procedure RecebeListaPedidosAmigos(Msg: string);
procedure RecebeListaArquivos(Msg: string);
procedure RecebeMsgConversa(Msg: string);
procedure RecebeNudge(Msg: string);
procedure RecebeDadosGerais(Msg: string);
procedure RecebeDadosServidor(Msg: string);
procedure AlteraDadosContato(Msg: string);
procedure AlteraStatusContato(Msg: string);
procedure ConexaoServidorMapaAceita(Msg: string);
procedure ConexaoServidorMapaNegada(Msg: string);
procedure AbreEdicaoPerfil(Msg: string);
procedure EncaminhaMsgProServidor(Msg: string);
procedure EncaminhaNudgeProServidor(Msg: string);
procedure EncaminhaEnvioArquivoProServidor(Msg: string);
procedure ConversaEstaPronto(Msg: string);
procedure EncaminhaExclusaoRegistroProServidor(Msg: string);
procedure PreencheListaUsuariosConectadosMapa(Msg: string);
procedure MovimentaUsuarioMapa(Msg: string);
procedure CriaUsuarioMapa(Msg: string);
procedure DestroiUsuarioMapa(Msg: string);
procedure SeDesconectaDoServidor(Msg: string);
```

```

    procedure RecebeDadosContato(Msg: string);
    procedure EnviaDadosContato(Msg: string);
    procedure AdicionaUsuarioPorChave(Msg: string);
    procedure SobreporServerClientRead(Sender: TObject; Socket:
TCustomWinSocket);
    procedure SobreporPanelClick(Sender: TObject);
    procedure SobreporClientMapError(Sender: TObject; Socket:
TCustomWinSocket;
        ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure SobreporClientMapConnect(Sender: TObject; Socket:
TCustomWinSocket);
    procedure SobreporClientMapDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
    procedure SobreporClientMapRead(Sender: TObject; Socket:
TCustomWinSocket);
    procedure VerificaMsgsEspera(ChaveContato: integer);
    procedure SeDesconectaDoServidorMapa(Msg: string);
    { Public declarations }
end;

var
    fServidores: TfServidores;

implementation

uses uDtModuleClient;

{$R *.dfm}
{$R Resources.RES}

// -----

```



```
// Ações dos Botões
```

```
// -----
```

```
procedure TfServidores.btConectarClick(Sender: TObject);
begin
  if (FrameSelecioneado>-1) then
  begin
    if (not
fileexists(extractfilepath(application.ExeName)+'Maps\' +Servidor[FrameSelecioneado].Mapa.NomeArquivoMapa)) then
    begin
      if (Application.MessageBox(PChar('Não foi encontrado o mapa
'+extractfilepath(application.ExeName)+'Maps\' +Servidor[FrameSelecioneado].Mapa.NomeArquivoMapa+' em seu computador. Deseja abrir a página de downloads de mapas para baixá-lo?'), 'Mapa não encontrado', 4)=idYES) then
        ShellExecute(Application.Handle, nil,
PChar('http://www.3dblah.com.br/downloads/mapas'), nil, nil,
SW_SHOWNORMAL);
      end
    else if (not
fileexists(extractfilepath(application.ExeName)+'Textures\' +Servidor[FrameSelecioneado].Mapa.NomeArquivoTextura)) then
    begin
      if (Application.MessageBox(PChar('Não foi encontrado o arquivo de texturas
'+extractfilepath(application.ExeName)+'Textures\' +Servidor[FrameSelecioneado].Mapa.NomeArquivoTextura+' em seu computador. Deseja abrir a página de downloads de mapas para baixá-lo?'), 'Mapa não encontrado', 4)=idYES) then
        ShellExecute(Application.Handle, nil,
PChar('http://www.3dblah.com.br/downloads/mapas'), nil, nil,
SW_SHOWNORMAL);
      end
    end
  end
end
```

```

    else
    begin
        StatusBar.Panels.Items[0].Text := ' Conectando-se ao mapa...';
        DtModuleClient.ClientMap.Host :=
Servidor[FrameSelecionado].Mapa.HostName;
        DtModuleClient.ClientMap.Open;
    end;
end
else
    showmessage('Por favor, selecione um servidor!');
end;

procedure TfServidores.btAtualizarClick(Sender: TObject);
begin
    SolicitaDadosGerais();
end;

procedure TfServidores.btPerfilClick(Sender: TObject);
var Msg: string;
begin
    Msg := '007';
    DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;

procedure TfServidores.btConfigClick(Sender: TObject);
begin
    fConfiguracoes := TfConfiguracoes.Create(nil);
    fConfiguracoes.ShowModal;
    freeandnil(fConfiguracoes);
end;

```

```

procedure TfServidores.btLogoffClick(Sender: TObject);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('019-1')); //Altera status para
OffLine
    modalresult := mrok;
end;

```

```

procedure TfServidores.btSairClick(Sender: TObject);
begin
    if (Application.MessageBox('    Tem certeza que deseja fechar o 3D BLAH?    ',
'Fechar Aplicação', 4)=idYES) then
        application.Terminate;
end;

```

```

procedure TfServidores.btEsquerdaClick(Sender: TObject);
begin
    dec(PaginaServidores);
    SolicitaPaginaServidores(PaginaServidores);
    if (PaginaServidores=1) then
        begin
            btEsquerda.Enabled := false;
        end;
        btDireita.Enabled := true;
end;

```

```

procedure TfServidores.btDireitaClick(Sender: TObject);
begin
    inc(PaginaServidores);
    SolicitaPaginaServidores(PaginaServidores);
    if (PaginaServidores=TotalPaginasServidores) then
        begin

```

```

        btDireita.Enabled := false;
    end;
    btEsquerda.Enabled := true;
end;

// -----
//  Fim das Ações dos Botões
// -----

// -----
//  Trata as mensagens recebidas do servidor principal
// -----

procedure TfServidores.ClientReadLogin(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Lista: TStringList;
    x: integer;
begin
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x], Codigo, Msg);
                if (Codigo=006) then
                    AlteraDadosContato(Msg)
                else if (Codigo=008) then
                    AbreEdicaoPerfil(Msg)
                else if (Codigo=010) then

```

```

        PreencheContatosNomes(Msg)
    else if (Codigo=014) then
        AmigoAdicionado(Msg)
    else if (Codigo=017) then
        RecebeListaPedidosAmigos(Msg)
    else if (Codigo=018) then
        RecebeDadosContato(Msg)
    else if (Codigo=020) then
        AlteraStatusContato(Msg)
    else if (Codigo=027) then
        RecebeMsgConversa(Msg)
    else if (Codigo=031) then
        RecebeDadosServidor(Msg)
    else if (Codigo=036) then
        RecebeDadosGerais(Msg)
    else if (Codigo=048) then
        SeDesconectaDoServidor(Msg)
    else if (Codigo=051) then
        RecebeNudge(Msg)
    else if (Codigo=060) then
        RecebeListaArquivos(Msg);
    end;
finally
    freeandnil(Lista);
end;
end;

procedure TfServidores.AlterarDadosContato(Msg: string);
var Lista: TStringList;
    Indice: integer;
    Posicao: integer;

```

```

    x: integer;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    Indice := frListaContatos.IndexOf(strtoint(Lista[0]));
    frListaContatos.Contatos[Indice].nome := Lista.Strings[1];
    frListaContatos.Contatos[Indice].apelido := Lista.Strings[2];
    frListaContatos.Contatos[Indice].nascimento := strtodate(Lista.Strings[3]);
    frListaContatos.Contatos[Indice].sexo := strtoint(Lista.Strings[4]);
    frListaContatos.Contatos[Indice].estado_civil := strtoint(Lista.Strings[5]);
    frListaContatos.Contatos[Indice].cidade := Lista.Strings[6];
    frListaContatos.Contatos[Indice].estado := Lista.Strings[7];
    frListaContatos.Contatos[Indice].pais := strtoint(Lista.Strings[8]);
    frListaContatos.Contatos[Indice].telefone := Lista.Strings[9];
    frListaContatos.Contatos[Indice].celular := Lista.Strings[10];
    frListaContatos.Contatos[Indice].status := strtoint(Lista.Strings[11]);
    frListaContatos.Contatos[Indice].modelo.chave_modelo :=
    strtoint(Lista.Strings[12]);
    Posicao := frListaContatos.Contatos[Indice].posicao;

    Indice := posicao-1;
    if (Posicao>1) and
    (UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao-
    1)].apelido)>UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posi
    cao)].apelido)) then
    begin
        TStringGridHack(frListaContatos.Grid).DeleteRow(Posicao-1);
        x := Posicao-1;
        while (x>0) and
        (UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(x)].apelido)>Up

```

```

perCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].apelido))
do
    dec(x);
    TStringGridHack(frListaContatos.Grid).InsertRow(x);
    frListaContatos.ReOrganizaPosicoes(Posicao,x+1);
    Indice := x;
end
else if (frListaContatos.ContatoCount>Posicao) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao+1)].ape
lido)<UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].a
pelido)) then
begin
    TStringGridHack(frListaContatos.Grid).DeleteRow(Posicao-1);
    x := Posicao+1;
    while (x<frListaContatos.ContatoCount) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(x)].apelido)<Up
perCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].apelido))
do
        inc(x);
        TStringGridHack(frListaContatos.Grid).InsertRow(x-1);
        frListaContatos.ReOrganizaPosicoes(Posicao,x);
        Indice := x-1;
end;
frListaContatos.Grid.Cells[2,Indice] := Lista.Strings[2];
frListaContatos.Grid.Cells[3,Indice] := Status(strtoint(Lista.Strings[11]));

Indice := DtModuleClient.ChavesUsuarios.IndexOf(Lista[0]);
if (Indice<>-1) then

DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo('023'+Lista
[2]));

```

```

    freeandnil(Lista);
end;

procedure TfServidores.AbreEdicaoPerfil(Msg: string);
var x: integer;
begin
    fPerfil := TfPerfil.Create(nil);
    fPerfil.Usuario := Usuario;
    fPerfil.PreencheTelaDadosPessoais();
    fPerfil.PreencheListaAvatares(Msg);
    fPerfil.PreencheTelaAvatar();
    if (fPerfil.ShowModal=mrok) then
    begin
        fPerfil.PreencheClasse();
        Usuario := fPerfil.Usuario;
        frListaContatos.lbNickUsuario.Caption := Usuario.apelido;
        if (Usuario.foto<>nil) then
            frListaContatos.imgUsuario.Picture := Usuario.foto;
        frListaContatos.cbStatus.ItemIndex := Usuario.status;
        msg := '005'+inttostr(Usuario.chave_usuario)+separador+
            Usuario.nome+separador+
            Usuario.apelido+separador+
            datetostr(Usuario.nascimento)+separador+
            inttostr(Usuario.sexo)+separador+
            inttostr(Usuario.estado_civil)+separador+
            Usuario.cidade+separador+
            Usuario.estado+separador+
            inttostr(Usuario.pais)+separador+
            Usuario.telefone+separador+
            Usuario.celular+separador+

```



```

        inttostr(Usuario.status)+separador+
        inttostr(Usuario.modelo.chave_modelo);
    DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
    Msg := '022'+Usuario.apelido;
    for x := 0 to DtModuleClient.Server.Socket.ActiveConnections-1 do
        DtModuleClient.Server.Socket.Connections[x].SendText(Protocolo(Msg));
    end;
    fPerfil.frAvatar.EngineEnd;
    freeandnil(fPerfil);
end;

procedure TfServidores.PreencheContatosNomes(Msg: string);
var ListaNomes: TStringList;
    ListaChaves: TStringList;
    ListaStatus: TStringList;
    ListaSexos: TStringList;
    ListaEmails: TStringList;
    ListaPaises: TStringList;
    x: integer;
begin
    if (Msg<>") then
        begin
            ListaNomes := TStringList.Create;
            ListaChaves := TStringList.Create;
            ListaStatus := TStringList.Create;
            ListaSexos := TStringList.Create;
            ListaEmails := TStringList.Create;
            ListaPaises := TStringList.Create;
            SeparaString6(Msg, ListaNomes, ListaChaves, ListaStatus, ListaSexos,
            ListaEmails, ListaPaises);
            frListaContatos.Grid.RowCount := ListaNomes.Count;

```

```

for x := 0 to ListaNomes.Count-1 do
begin
    frListaContatos.NovoContato();
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].chave_usuario :=
strtoint(ListaChaves[x]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].apelido :=
ListaNomes[x];
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].status :=
strtoint(ListaStatus[x]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].sexo :=
strtoint(ListaSexos[x]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].posicao :=
frListaContatos.ContatoCount;
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].email :=
ListaEmails[x];
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].pais :=
strtoint(ListaPaises[x]);
    frListaContatos.Grid.Cells[2,x] := ListaNomes[x];
    frListaContatos.Grid.Cells[3,x] := Status(strtoint(ListaStatus[x]));
end;
freeandnil(ListaNomes);
freeandnil(ListaChaves);
freeandnil(ListaStatus);
freeandnil(ListaSexos);
freeandnil(ListaEmails);
freeandnil(ListaPaises);
end;
end;

procedure TfServidores.AmigoAdicionado(Msg: string);
var Lista: TStringList;

```

```

Indice: integer;
Posicao: integer;
x: integer;
begin
  Lista := TStringList.Create;
  SeparaString(Msg, Lista);
  if (frListaContatos.IndexOf(strtoint(Lista[0]))=-1) then
  begin
    frListaContatos.NovoContato();
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].chave_usuario :=
strtoint(Lista[0]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].apelido := Lista[1];
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].status :=
strtoint(Lista[2]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].sexo :=
strtoint(Lista[3]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].pais :=
strtoint(Lista[4]);
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].email := Lista[5];
    frListaContatos.Contatos[frListaContatos.ContatoCount-1].posicao :=
frListaContatos.ContatoCount;
    Posicao := frListaContatos.Contatos[frListaContatos.ContatoCount-1].posicao;
    Indice := posicao-1;

    if (frListaContatos.ContatoCount>1) then
      frListaContatos.Grid.RowCount := frListaContatos.Grid.RowCount+1;

    if (Posicao>1) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao-
1)].apelido)>UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posi
cao)].apelido)) then

```

```

begin
  TStringGridHack(frListaContatos.Grid).DeleteRow(Posicao-1);
  x := Posicao-1;
  while (x>0) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(x)].apelido)>Up
perCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].apelido))
do
  dec(x);
  TStringGridHack(frListaContatos.Grid).InsertRow(x);
  frListaContatos.ReOrganizaPosicoes(Posicao,x+1);
  Indice := x;
end
else if (frListaContatos.ContatoCount>Posicao) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao+1)].ape
lido)<UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].a
pelido)) then
begin
  TStringGridHack(frListaContatos.Grid).DeleteRow(Posicao-1);
  x := Posicao+1;
  while (x<frListaContatos.ContatoCount) and
(UpperCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(x)].apelido)<Up
perCase(frListaContatos.Contatos[frListaContatos.IndexOfPos(Posicao)].apelido))
do
  inc(x);
  TStringGridHack(frListaContatos.Grid).InsertRow(x-1);
  frListaContatos.ReOrganizaPosicoes(Posicao,x);
  Indice := x-1;
end;
frListaContatos.Grid.Cells[2,Indice] := Lista[1];
frListaContatos.Grid.Cells[3,Indice] := Status(strtoint(Lista[2]));

```

```

Indice := DtModuleClient.ChavesUsuarios.IndexOf(Lista[0]);
if (Indice<>-1) then
    DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo('039'));
end;
freeandnil(Lista);
end;

```

```

procedure TfServidores.RecebeListaPedidosAmigos(Msg: string);
var ListaChaves: TStringList;
    ListaNomes: TStringList;
    ListaEmails: TStringList;
    x: integer;
begin
    ListaChaves := TStringList.Create;
    ListaNomes := TStringList.Create;
    ListaEmails := TStringList.Create;
    SeparaString3(Msg, ListaChaves, ListaNomes, ListaEmails);
    for x := 0 to ListaChaves.Count-1 do
        begin
            if (Application.MessageBox(PAnsiChar(' Um usuário lhe enviou uma solicitação
de amigo. '+#13+
                ' Nome: '+ListaNomes[x]+#13+
                ' E-mail: '+ListaEmails[x]+#13+
                ' Deseja aceitar?'), 'Solicitação de Amigo', 4)=idYES) then
                begin
                    DtModuleClient.Client.Socket.SendText(Protocolo('012'+ListaEmails[x]));
                end
            else
                begin
                    DtModuleClient.Client.Socket.SendText(Protocolo('013'+ListaChaves[x]));
                end
            end;
        end
    end;
end;

```

```
end;  
freeandnil(ListaChaves);  
freeandnil(ListaNomes);  
freeandnil(ListaEmails);  
end;  
  
procedure TfServidores.RecebeDadosContato(Msg: string);  
var Lista: TStringList;  
    Indice: integer;  
    IndiceClientes: integer;  
    User: TUsuario;  
begin  
    Lista := TStringList.Create;  
    SeparaString(Msg, Lista);  
    Indice := frListaContatos.IndexOf(strtoint(Lista[0]));  
    if (Indice<>-1) then  
        frListaContatos.Contatos[Indice].PreencheClasse(Lista);  
  
        User := TUsuario.Create;  
        User.PreencheClasse(Lista);  
        with frListaContatos do  
            begin  
                IndiceClientes := DtModuleClient.ChavesUsuarios.IndexOf(Msg);  
                if (IndiceClientes<>-1) then  
  
DtModuleClient.Server.Socket.Connections[IndiceClientes].SendText(Protocolo('02  
4'+inttostr(User.chave_usuario)+separador+  
                    User.nome+separador+  
                    User.apelido+separador+  
                    datetostr(User.nascimento)+separador+  
                    inttostr(User.sexo)+separador+
```

```

        inttostr(User.estado_civil)+separador+
        User.cidade+separador+
        User.estado+separador+
        inttostr(User.pais)+separador+
        User.telefone+separador+
        User.celular+separador+
        inttostr(User.status)+separador+
        inttostr(User.Modelo.chave_modelo)+separador+
        User.Modelo.nome_arquivo+separador+
        User.email));

    end;
    freeandnil(User);
    freeandnil(Lista);
end;

procedure TfServidores.AlterarStatusContato(Msg: string);
var Lista: TStringList;
    linha: integer;
    Indice: integer;
    Txt: string;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    linha := -1;
    Indice := frListaContatos.IndexOf(strtoint(Lista[0]));
    if (Indice>-1) then
        linha := frListaContatos.Contatos[Indice].posicao-1;
    if (linha>-1) then
        begin
            if (frListaContatos.Contatos[Indice].status=-1) and (strtoint(Lista[1])<>-1) then
                begin

```

```

MsnPopup.ClosePopUps;
MsnPopup.Title := '';
MsnPopup.Text := frListaContatos.Contatos[Indice].apelido+' acabou de
entrar.';
MsnPopup.URL := frListaContatos.Contatos[Indice].apelido;
MsnPopup.Tag := frListaContatos.Contatos[Indice].chave_usuario;
if (configuracoes.SomLogin) then
    PlaySound(PChar(extractfilepath(application.ExeName)+'Sounds\login.wav'),
1, SND_ASYNC);
MsnPopup.ShowPopUp;
end;
frListaContatos.Contatos[Indice].status := strtoint(Lista[1]);
frListaContatos.Grid.Cells[3,linha] := Status(strtoint(Lista[1]));

if (strtoint(Lista[1])=-1) then
    Txt := '064'
else
    Txt := '065';

linha :=
DtModuleClient.ChavesUsuarios.IndexOf(inttostr(frListaContatos.Contatos[Indice].
chave_usuario));
if (linha<>-1) then
    if (DtModuleClient.ChavesUsuarios[linha]<>'EM ESPERA') then
        DtModuleClient.Server.Socket.Connections[linha].SendText(Protocolo(Txt))
    else
        begin

DtModuleClient.ChavesMsgsEmEspera.Add(inttostr(frListaContatos.Contatos[Indic
e].chave_usuario));
        DtModuleClient.MsgsEmEspera.Add(Txt);

```



```

        end;
    end;
    freeandnil(Lista);
end;

procedure TfServidores.RecebeMsgConversa(Msg: string);
var Lista: TStringList;
    ChaveContato: integer;
    ContatoApelido: string;
    Texto: string;
    Indice: integer;
    Skin: string;
    Amigo: string;
    SomNudge: string;
    SomMsg: string;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    ChaveContato := strtoint(Lista[0]);
    Texto := Lista[2]+separador+Lista[3];
    Indice := DtModuleClient.ChavesUsuarios.IndexOf(inttostr(ChaveContato));
    if (Indice<>-1) and (DtModuleClient.EstadosUsuarios[Indice]<>'EM ESPERA')
then
    DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo('028'+Text
o))
    else
    begin
        DtModuleClient.ChavesMsgsEmEspera.Add(Lista[0]);
        DtModuleClient.MsgsEmEspera.Add('028'+Texto);
        if (Indice=-1) then

```

```

begin
  DtModuleClient.ChavesUsuarios.Add(inttostr(ChaveContato));
  DtModuleClient.EstadosUsuarios.Add('EM ESPERA');
  Indice := frListaContatos.IndexOf(ChaveContato);
  if (Indice<>-1) then
    begin
      ContatoApelido := frListaContatos.Contatos[Indice].apelido;
      Amigo := 'S';
    end
  else
    begin
      ContatoApelido := 'Usuário Desconhecido';
      Amigo := 'N';
    end;
  if (configuracoes.SomMsg) then
    SomMsg := 'S'
  else
    SomMsg := 'N';
  if (configuracoes.SomNudge) then
    SomNudge := 'S'
  else
    SomNudge := 'N';
  if (configuracoes.SkinDialogo) then
    Skin := 'S'
  else
    Skin := 'N';
  ExecuteProgram(extractfilepath(application.ExeName)+'Conversa.exe',Skin+'
'+SomMsg+' '+SomNudge+' '+Amigo+' '+inttostr(Usuario.chave_usuario)+'
'+TrocaCaracter(Usuario.apelido,' ',SubstitutoEspaco)+' '+inttostr(ChaveContato)+'
'+TrocaCaracter(ContatoApelido,' ',SubstitutoEspaco));
end;

```

```

end;
freeandnil(Lista);
end;

procedure TfServidores.RecebeDadosServidor(Msg: string);
var Lista: TStringList;
    Pag: integer;
    Frame: integer;
begin
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    Pag := strtoint(Lista[0]);
    if (Pag=PaginaServidores) then
    begin
        Frame := strtoint(Lista[1]);
        Servidor[Frame] := TfrMapa.Create(nil);
        Servidor[Frame].SetBounds(FrameLeft(Frame+1),FrameTop(Frame+1), 337,
116);
        Servidor[Frame].Parent := pnMapas;
        inc(Incrementavel);
        Servidor[Frame].Name := 'Mapa'+inttostr(Incrementavel);
        Servidor[Frame].Mapa.ChaveServidor := strtoint(Lista[2]);
        Servidor[Frame].lbTitulo.Caption := Lista[3];
        Servidor[Frame].Mapa.Titulo := Lista[3];
        Servidor[Frame].reDescricao.Text := Lista[4];
        Servidor[Frame].Mapa.Descricao := Lista[4];
        Servidor[Frame].lbQtd.Caption := Lista[5];
        Servidor[Frame].Mapa.NumMaxUsuarios := strtoint(Lista[5]);
        Servidor[Frame].Mapa.HostName := Lista[6];
        Servidor[Frame].Mapa.NomeArquivoMapa := Lista[7];
        Servidor[Frame].Mapa.NomeArquivoTextura := Lista[8];
    end;
end;

```

```

Servidor[Frame].Panel.OnClick := SobreporPanelClick;
Servidor[Frame].Image.OnClick := SobreporPanelClick;
Servidor[Frame].lbTitulo.OnClick := SobreporPanelClick;
Servidor[Frame].lbQtd.OnClick := SobreporPanelClick;
Servidor[Frame].Panel.OnDblClick := btConectarClick;
Servidor[Frame].Image.OnDblClick := btConectarClick;
Servidor[Frame].lbTitulo.OnDblClick := btConectarClick;
Servidor[Frame].lbQtd.OnDblClick := btConectarClick;
Servidor[Frame].CarregarImagem;
end;
freeandnil(Lista);
end;

```

```

procedure TfServidores.RecebeDadosGerais(Msg: string);
var Lista: TStringList;
begin
  Lista := TStringList.Create;
  SeparaString(Msg, Lista);
  TotalPaginasServidores := strtoint(Lista[0]);
  SolicitaPaginaServidores(PaginaServidores);
  if (PaginaServidores<TotalPaginasServidores) then
    btDireita.Enabled := true;
  freeandnil(Lista);
end;

```

```

procedure TfServidores.SeDesconectaDoServidor(Msg: string);
begin
  if (length(Msg)>0) then
    showmessage('Você foi desconectado do Servidor pelo administrador! Motivo:
'+Msg)
  else

```

```

    showmessage('Você foi desconectado do Servidor pelo administrador!');
    DtModuleClient.Client.Close;
end;

procedure TfServidores.RecebeNudge(Msg: string);
var ChaveContato: integer;
    ContatoApelido: string;
    Indice: integer;
    Skin: string;
    Amigo: string;
    SomNudge: string;
    SomMsg: string;
begin
    ChaveContato := strtoint(Msg);
    Indice := DtModuleClient.ChavesUsuarios.IndexOf(inttostr(ChaveContato));
    if (Indice<>-1) and (DtModuleClient.EstadosUsuarios[Indice]<>'EM ESPERA')
then
    DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo('052'))
else
begin
    DtModuleClient.ChavesMsgsEmEspera.Add(Msg);
    DtModuleClient.MsgsEmEspera.Add('052');
    if (Indice=-1) then
begin
    DtModuleClient.ChavesUsuarios.Add(inttostr(ChaveContato));
    DtModuleClient.EstadosUsuarios.Add('EM ESPERA');
    Indice := frListaContatos.IndexOf(ChaveContato);
    if (Indice<>-1) then
begin
    ContatoApelido := frListaContatos.Contatos[Indice].apelido;
    Amigo := 'S';

```

```

end
else
begin
    ContatoApelido := 'Usuário Desconhecido';
    Amigo := 'N';
end;
if (configuracoes.SomMsg) then
    SomMsg := 'S'
else
    SomMsg := 'N';
if (configuracoes.SomNudge) then
    SomNudge := 'S'
else
    SomNudge := 'N';
if (configuracoes.SkinDialogo) then
    Skin := 'S'
else
    Skin := 'N';
    ExecuteProgram(extractfilepath(application.ExeName)+'Conversa.exe',Skin+'
'+SomMsg+' '+SomNudge+' '+Amigo+' '+inttostr(Usuario.chave_usuario)+'
'+TrocaCaracter(Usuario.apelido,' ',SubstitutoEspaco)+' '+inttostr(ChaveContato)+'
'+TrocaCaracter(ContatoApelido,' ',SubstitutoEspaco));
end;
end;
end;

procedure TfServidores.RecebeListaArquivos(Msg: string);
var ListaChaves: TStringList;
    ListaNomes: TStringList;
    ListaEmails: TStringList;
    ListaNomeArquivos: TStringList;

```

```

Indice: integer;
x: integer;
ContatoApelido: string;
Skin: string;
Amigo: string;
SomMsg: string;
SomNudge: string;
begin
  ListaChaves := TStringList.Create;
  ListaNomes := TStringList.Create;
  ListaEmails := TStringList.Create;
  ListaNomeArquivos := TStringList.Create;
  SeparaString4(Msg, ListaChaves, ListaNomes, ListaEmails, ListaNomeArquivos);
  for x := 0 to ListaChaves.Count-1 do
    begin
      Indice := DtModuleClient.ChavesUsuarios.IndexOf(ListaChaves[x]);
      if (Indice<>-1) and (DtModuleClient.EstadosUsuarios[Indice]<>'EM ESPERA')
then
        begin

DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo('061'+Lista
NomeArquivos[x]));
        end
      else
        begin
          DtModuleClient.ChavesMsgsEmEspera.Add(ListaChaves[x]);
          DtModuleClient.MsgsEmEspera.Add('061'+ListaNomeArquivos[x]);
          if (Indice=-1) then
            begin
              DtModuleClient.ChavesUsuarios.Add(ListaChaves[x]);
              DtModuleClient.EstadosUsuarios.Add('EM ESPERA');
            end
          end
        end
      end
    end
  end
end

```

```

Indice := frListaContatos.IndexOf(strtoint(ListaChaves[x]));
if (Indice<>-1) then
begin
  ContatoApelido := frListaContatos.Contatos[Indice].apelido;
  Amigo := 'S';
end
else
begin
  ContatoApelido := 'Usuário Desconhecido';
  Amigo := 'N';
end;
if (configuracoes.SomMsg) then
  SomMsg := 'S'
else
  SomMsg := 'N';
if (configuracoes.SomNudge) then
  SomNudge := 'S'
else
  SomNudge := 'N';
if (configuracoes.SkinDialogo) then
  Skin := 'S'
else
  Skin := 'N';
  ExecuteProgram(extractfilepath(application.ExeName)+'Conversa.exe',Skin+'
'+SomMsg+' '+SomNudge+' '+Amigo+' '+inttostr(Usuario.chave_usuario)+'
'+TrocaCaracter(Usuario.apelido,' ',SubstitutoEspaco)+' '+ListaChaves[x]+'
'+TrocaCaracter(ContatoApelido,' ',SubstitutoEspaco));
end;
end;
end;
freeandnil(ListaChaves);

```



```

    freeandnil(ListaNomes);
    freeandnil(ListaEmails);
    freeandnil(ListaNomeArquivos);
end;

// -----
//  Fim das mensagens recebidas do servidor principal
// -----

// -----
//  Trata as mensagens recebidas do cliente Conversa.exe
// -----

procedure TfServidores.SobreporServerClientRead(Sender: TObject; Socket:
TCustomWinSocket);
var Codigo: integer;
    Msg: string;
    Lista: TStringList;
    x: integer;
begin
    Lista := TStringList.Create;
    try
        DivideMsgs(Socket.ReceiveText, Lista);
        for x := 0 to Lista.Count-1 do
            begin
                DecodificaString(Lista[x], Codigo, Msg);
                if (Codigo=004) then
                    AdicionaUsuarioPorChave(Msg)
                else if (Codigo=021) then
                    EnviaDadosContato(Msg)
                else if (Codigo=025) then

```

```

    EncaminhaMsgProServidor(Msg)
else if (Codigo=047) then
    SolicitaListaArquivos()
else if (Codigo=049) then
    EncaminhaNudgeProServidor(Msg)
else if (Codigo=055) then
    EncaminhaEnvioArquivoProServidor(Msg)
else if (Codigo=057) then
    ConversaEstaPronto(Msg)
else if (Codigo=062) then
    EncaminhaExclusaoRegistroProServidor(Msg);
end;
finally
    freeandnil(Lista);
end;
end;

procedure TfServidores.AdicionaUsuarioPorChave(Msg: string);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('011'+Msg));
end;

procedure TfServidores.EnviaDadosContato(Msg: string);
var Indice: integer;
    IndiceClientes: integer;
begin
    Indice := frListaContatos.IndexOf(strtoint(Msg));
    if (Indice<>-1) then
        with frListaContatos do
            begin
                IndiceClientes := DtModuleClient.ChavesUsuarios.IndexOf(Msg);

```

```

if (IndiceClientes<>-1) then

DtModuleClient.Server.Socket.Connections[IndiceClientes].SendText(Protocolo('02
4'+inttostr(Contatos[Indice].chave_usuario)+separador+
                Contatos[Indice].nome+separador+
                Contatos[Indice].apelido+separador+
                datetostr(Contatos[Indice].nascimento)+separador+
                inttostr(Contatos[Indice].sexo)+separador+
                inttostr(Contatos[Indice].estado_civil)+separador+
                Contatos[Indice].cidade+separador+
                Contatos[Indice].estado+separador+
                inttostr(Contatos[Indice].pais)+separador+
                Contatos[Indice].telefone+separador+
                Contatos[Indice].celular+separador+
                inttostr(Contatos[Indice].status)+separador+

inttostr(Contatos[Indice].Modelo.chave_modelo)+separador+

Contatos[Indice].Modelo.nome_arquivo+separador+
                Contatos[Indice].email));

    end;

    DtModuleClient.Client.Socket.SendText(Protocolo('015'+Msg))
end;

procedure TfServidores.EncaminhaMsgProServidor(Msg: string);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('026'+Msg));
end;

procedure TfServidores.SolicitaListaArquivos;
begin

```

```

    DtModuleClient.Client.Socket.SendText(Protocolo('059'));
end;

procedure TfServidores.EncaminhaNudgeProServidor(Msg: string);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('050'+Msg));
end;

procedure TfServidores.EncaminhaEnvioArquivoProServidor(Msg: string);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('056'+Msg));
end;

procedure TfServidores.ConversaEstaPronto(Msg: string);
var Indice: integer;
begin
    Indice := DtModuleClient.ChavesUsuarios.IndexOf(Msg);
    if (Indice<>-1) then
    begin
        DtModuleClient.EstadosUsuarios[Indice] := 'PRONTO';
        VerificaMsgsEspera(strtoint(Msg));
    end;
end;

procedure TfServidores.EncaminhaExclusaoRegistroProServidor(Msg: string);
begin
    DtModuleClient.Client.Socket.SendText(Protocolo('063'+Msg));
end;

// -----
// Fim das mensagens recebidas do cliente Conversa.exe

```

```
// -----
```

```
// -----
```

```
// Trata as mensagens recebidas do servidor de Mapa
```

```
// -----
```

```
procedure TfServidores.SobreporClientMapRead(Sender: TObject; Socket:
TCustomWinSocket);
```

```
var Codigo: integer;
```

```
    Msg: string;
```

```
    Lista: TStringList;
```

```
    x: integer;
```

```
begin
```

```
    Lista := TStringList.Create;
```

```
    try
```

```
        DivideMsgs(Socket.ReceiveText, Lista);
```

```
        for x := 0 to Lista.Count-1 do
```

```
            begin
```

```
                DecodificaString(Lista[x], Codigo, Msg);
```

```
                if (Codigo=033) then
```

```
                    ConexaoServidorMapaAceita(Msg)
```

```
                else if (Codigo=034) then
```

```
                    ConexaoServidorMapaNegada(Msg)
```

```
                else if (Codigo=038) then
```

```
                    PreencheListaUsuariosConectadosMapa(Msg)
```

```
                else if (Codigo=043) then
```

```
                    CriaUsuarioMapa(Msg)
```

```
                else if (Codigo=044) then
```

```
                    DestroiUsuarioMapa(Msg)
```

```
                else if (Codigo=046) then
```

```
                    MovimentaUsuarioMapa(Msg)
```

```

    else if (Codigo=058) then
        SeDesconectaDoServidorMapa(Msg);
    end;
finally
    freeandnil(Lista);
end;
end;

procedure TfServidores.ConexaoServidorMapaAceita(Msg: string);
begin
    if (fPrincipal=nil) then
        begin
            fPrincipal := TfPrincipal.Create(nil);
            fPrincipal.Mapa.NomeArquivoMapa :=
extractfilepath(application.ExeName)+'Maps\' +Servidor[FrameSelecionado].Mapa.
NomeArquivoMapa;
            fPrincipal.Mapa.NomeArquivoTextura :=
extractfilepath(application.ExeName)+'Textures\' +Servidor[FrameSelecionado].Ma
pa.NomeArquivoTextura;
            fPrincipal.Mapa.Titulo := Servidor[FrameSelecionado].Mapa.Titulo;
            fPrincipal.EngineInit();
            frListaContatos.Parent := fPrincipal.pnListaContatos;
            frListaContatos.Grid.OnClick := fPrincipal.PerdeFoco;
            frListaContatos.pnBotoes.OnClick := fPrincipal.PerdeFoco;
            frListaContatos.Panel.OnClick := fPrincipal.PerdeFoco;
            Application.OnIdle := fPrincipal.ApplIdle;
            StatusBar.Panels.Items[0].Text := "";
            fPrincipal.OnKeyDown := FormKeyDown;
            fPrincipal.ShowModal;
            frListaContatos.cbStatus.Enabled := true;
            Application.OnIdle := nil;

```

```

fPrincipal.PerdeFoco(nil);
frListaContatos.Parent := pnListaContatos;
frListaContatos.Grid.OnClick := nil;
frListaContatos.pnBotoes.OnClick := nil;
frListaContatos.Panel.OnClick := nil;
freeandnil(fPrincipal);
DtModuleCLient.ClientMap.Close;
end;
end;

```

```

procedure TfServidores.ConexaoServidorMapaNegada(Msg: string);
begin
    DtModuleClient.ClientMap.Close;
    showmessage('Não foi possível se conectar a este servidor:'+#13+Msg);
    StatusBar.Panels.Items[0].Text := '';
end;

```

```

procedure TfServidores.PreencheListaUsuariosConectadosMapa(Msg: string);
var ChavesUsuarios: TStringList;
    ArquivosModelos: TStringList;
    Apelidos: TStringList;
    PosicaoX: TStringList;
    PosicaoY: TStringList;
    PosicaoZ: TStringList;
    RotacaoY: TStringList;
    x: integer;
begin
    if (fPrincipal<>nil) and (Msg<>") then
    begin
        ChavesUsuarios := TStringList.Create;
        ArquivosModelos := TStringList.Create;

```

```

Apelidos := TStringList.Create;
PosicaoX := TStringList.Create;
PosicaoY := TStringList.Create;
PosicaoZ := TStringList.Create;
RotacaoY := TStringList.Create;
SeparaString7(Msg, ChavesUsuarios, ArquivosModelos, Apelidos, PosicaoX,
PosicaoY, PosicaoZ, RotacaoY);
setlength(fPrincipal.Usuarios, ChavesUsuarios.Count);
for x := 0 to ChavesUsuarios.Count-1 do
begin
    fPrincipal.Usuarios[x] := TUsuarioConectado.Create;
    fPrincipal.Usuarios[x].Chave_Usuario := strtoint(ChavesUsuarios[x]);
    fPrincipal.Usuarios[x].Apelido := Apelidos[x];
    fPrincipal.Usuarios[x].Modelo.nome_arquivo := ArquivosModelos[x];
    fPrincipal.Usuarios[x].LoadAvatar;
    fPrincipal.Usuarios[x].Personagem.SetPosition(strtfloat(PosicaoX[x]),
strtfloat(PosicaoY[x]), strtfloat(PosicaoZ[x]));
    fPrincipal.Usuarios[x].Personagem.SetRotation(0,strtfloat(RotacaoY[x]),0);
end;
freeandnil(ChavesUsuarios);
freeandnil(ArquivosModelos);
freeandnil(Apelidos);
freeandnil(PosicaoX);
freeandnil(PosicaoY);
freeandnil(PosicaoZ);
freeandnil(RotacaoY);
end;
end;

procedure TfServidores.CriaUsuarioMapa(Msg: string);
var Lista: TStringList;

```



```

x: integer;
JaExiste: boolean;
begin
  if (fPrincipal<>nil) and (Msg<>") then
  begin
    JaExiste := false;
    Lista := TStringList.Create;
    SeparaString(Msg, Lista);
    for x := 0 to length(fPrincipal.Usuarios)-1 do
      if (fPrincipal.Usuarios[x]<>nil) and
(fPrincipal.Usuarios[x].Chave_Usuario=strtoint(Lista[0])) then
        begin
          JaExiste := true;
          break;
        end;
    if (not JaExiste) then
    begin
      x := length(fPrincipal.Usuarios)+1;
      setlength(fPrincipal.Usuarios, x);
      dec(x);
      fPrincipal.Usuarios[x] := TUsuarioConectado.Create;
      fPrincipal.Usuarios[x].Chave_Usuario := strtoint(Lista[0]);
      fPrincipal.Usuarios[x].Apelido := Lista[2];
      fPrincipal.Usuarios[x].Modelo.nome_arquivo := Lista[1];
      fPrincipal.Usuarios[x].LoadAvatar;
      fPrincipal.Usuarios[x].Personagem.SetPosition(strtofloat(Lista[3]),
strtofloat(Lista[4]), strtofloat(Lista[5]));
      fPrincipal.Usuarios[x].Personagem.SetRotation(0,strtofloat(Lista[6]),0);
    end;
    freeandnil(Lista);
  end;
end;

```

end;

```
procedure TfServidores.DestroyUsuarioMapa(Msg: string);
```

```
begin
```

```
  if (fPrincipal<>nil) then
```

```
    fPrincipal.DestroyUsuarioMapa(strtoint(Msg));
```

```
end;
```

```
procedure TfServidores.MovimentaUsuarioMapa(Msg: string);
```

```
var Lista: TStringList;
```

```
  x: integer;
```

```
begin
```

```
  if (fPrincipal<>nil) then
```

```
  begin
```

```
    Lista := TStringList.Create;
```

```
    try
```

```
      SeparaString(Msg, Lista);
```

```
      for x := 0 to length(fPrincipal.Usuarios)-1 do
```

```
        begin
```

```
          if (fPrincipal.Usuarios[x]<>nil) and
```

```
(fPrincipal.Usuarios[x].Chave_Usuario=strtoint(Lista[0])) then
```

```
          begin
```

```
fPrincipal.Usuarios[x].Personagem.SetPosition(strtfloat(Lista[1]),strtfloat(Lista[2])
,strtfloat(Lista[3]));
```

```
fPrincipal.Usuarios[x].Personagem.SetRotation(0,(radtodeg(strtfloat(Lista[4]))+18
0)*-1,0);
```

```
      break;
```

```
    end;
```

```
  end;
```

```

    finally
        freeandnil(Lista);
    end;
end;
end;

procedure TfServidores.SeDesconectaDoServidorMapa(Msg: string);
begin
    if (length(Msg)>0) then
        showmessage('Você foi desconectado do Servidor de Mapas pelo
administrador! Motivo: '+Msg)
    else
        showmessage('Você foi desconectado do Servidor de Mapas pelo
administrador!');
        DtModuleClient.ClientMap.Close;
    end;

    // -----
    //  Fim das mensagens recebidas do servidor de Mapa
    // -----

    // -----
    //  Eventos dos componentes
    // -----

    procedure TfServidores.FormCreate(Sender: TObject);
    begin
        try
            DtModuleClient.Server.Active := true;
        except
        end;
    end;

```

```

DtModuleClient.Client.OnRead := ClientReadLogin;
DtModuleClient.Server.OnClientRead := SobreporServerClientRead;
DtModuleClient.ClientMap.OnConnect := SobreporClientMapConnect;
DtModuleClient.ClientMap.OnError := SobreporClientMapError;
DtModuleClient.ClientMap.OnDisconnect := SobreporClientMapDisconnect;
DtModuleClient.ClientMap.OnRead := SobreporClientMapRead;
Usuario := TUsuario.Create;
MsnPopup := TMsnPopup.Create(nil);
MsnPopup.BackgroundImage.LoadFromResourceID(hInstance,1);
PaginaServidores := 1;
TotalPaginasServidores := 1;
Incrementavel := 1;
FrameSelecionado := -1;
SolicitaDadosGerais();
SolicitaListaContatos();
SolicitaListaPedidos();
SolicitaListaArquivos();
end;

procedure TfServidores.FormDestroy(Sender: TObject);
var x: integer;
begin
  freeandnil(Usuario);
  freeandnil(MsnPopup);
  for x := 0 to 7 do
    freeandnil(Servidor[x]);
  for x := 0 to frListaContatos.ContatoCount-1 do
    freeandnil(frListaContatos.Contatos[x]);
end;

```

```
procedure TfServidores.FormKeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
```

```
begin
```

```
  if (ssCtrl in Shift) and (chr(Key) in ['Y', 'y']) then
```

```
  begin
```

```
    if (fConsole=nil) then
```

```
    begin
```

```
      try
```

```
        freeandnil(fConsole);
```

```
        fConsole := TfConsole.Create(application);
```

```
        fConsole.Console();
```

```
      finally
```

```
        freeandnil(fConsole);
```

```
      end;
```

```
    end;
```

```
  end;
```

```
end;
```

```
procedure TfServidores.MSNPopUpClick(Sender: TObject);
```

```
begin
```

```
  frListaContatos.IniciarConversa(Usuario.chave_usuario, Usuario.apelido,(Sender
as TMsnPopup).Tag,(Sender as TMsnPopup).URL);
```

```
end;
```

```
procedure TfServidores.frListaContatoscbStatusChange(Sender: TObject);
```

```
begin
```

```
  Usuario.status := frListaContatos.cbStatus.ItemIndex;
```

```
  DtModuleClient.Client.Socket.SendText(Protocolo('019'+inttostr(Usuario.status)));
```

```
end;
```

```
procedure TfServidores.frListaContatosGridDbClick(Sender: TObject);
```

```

begin
    if
(frListaContatos.Contatos[frListaContatos.IndexOfPos(frListaContatos.Grid.Row+1)
].status=-1) then
        showmessage('Este usuário não está conectado no momento!')
    else
        begin

frListaContatos.IniciarConversa(Usuario.chave_usuario,Usuario.apelido,frListaCon
tatos.Contatos[frListaContatos.IndexOfPos(frListaContatos.Grid.Row+1)].chave_us
uario,frListaContatos.Contatos[frListaContatos.IndexOfPos(frListaContatos.Grid.Ro
w+1)].apelido);
        end;
    end;

procedure TfServidores.frListaContatosGridKeyDown(Sender: TObject; var Key:
Word; Shift: TShiftState);
begin
    Key := 0;
end;

// -----
//  Fim dos eventos dos componentes
// -----

// -----
//  Outras funções
// -----

procedure TfServidores.PreencheClasseUsuario(Msg: string);
var Lista: TStringList;

```

```

begin
  Lista := TStringList.Create;
  try
    SeparaString(Msg, Lista);
    Usuario.PreencheClasse(Lista);
    Usuario.CarregarImagem(frListaContatos.imgUsuario);
  finally
    freeandnil(Lista);
  end;
end;

```

```

procedure TfServidores.SolicitaListaContatos;
var Msg: string;
begin
  Msg := '009';
  DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;

```

```

procedure TfServidores.SolicitaListaPedidos;
var Msg: string;
begin
  Msg := '016';
  DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;

```

```

procedure TfServidores.SolicitaPaginaServidores(Pagina: integer);
var Msg: string;
    x: integer;
begin
  Msg := '030'+inttostr(Pagina);
  for x := 0 to 7 do

```

```

    freeandnil(Servidor[x]);
    FrameSelecionado := -1;
    DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;

procedure TfServidores.SolicitaDadosGerais;
var Msg: string;
begin
    Msg := '035';
    DtModuleClient.Client.Socket.SendText(Protocolo(Msg));
end;

procedure TfServidores.SobreporPanelClick(Sender: TObject);
var x: integer;
begin
    for x := 0 to 7 do
        if (Servidor[x]<>nil) then
            Servidor[x].DeSelecionaFrame;
        if (Sender is TPanel) then
            ((Sender as TPanel).Owner as TfrMapa).SelecionaFrame()
        else if (Sender is TImage) then
            ((Sender as TImage).Owner as TfrMapa).SelecionaFrame()
        else if (Sender is TLabel) then
            ((Sender as TLabel).Owner as TfrMapa).SelecionaFrame();
    for x := 0 to 7 do
        if (Servidor[x].Selecionado) then
            begin
                FrameSelecionado := x;
                break;
            end;
    end;
end;

```



```

procedure TfServidores.SobreporClientMapError(Sender: TObject; Socket:
TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  if (ErrorCode=10061) then
  begin
    StatusBar.Panels.Items[0].Text := '';
    showmessage('Não foi possível se conectar a este servidor. Por favor, escolha
um outro servidor ou tente novamente mais tarde!');
    ErrorCode := 0;
  end
  else
  begin
    showmessage('Erro '+inttostr(ErrorCode));
    ErrorCode := 0;
  end;
end;

```

```

procedure TfServidores.SobreporClientMapConnect(Sender: TObject; Socket:
TCustomWinSocket);
var Txt: string;
begin
  StatusBar.Panels.Items[0].Text := ' Recebendo informações do mapa...';
  Txt :=
inttostr(Usuario.chave_usuario)+separador+Usuario.modelo.nome_arquivo+separ
ador+Usuario.apelido+

separador+inttostr(Servidor[FrameSelecionado].Mapa.ChaveServidor)+separador+
inttostr(FileSize(extractfilepath(application.ExeName)+'Maps\' +Servidor[FrameSele
cionado].Mapa.NomeArquivoMapa));

```

```

    DtModuleClient.ClientMap.Socket.SendText(Protocolo('032'+Txt));
end;

procedure TfServidores.SobrepClientMapDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
begin
    if (fPrincipal<>nil) then
    begin
        fPrincipal.ModalResult := mrcancel;
        showmessage('A conexão com o mapa foi perdida!');
    end;
end;

procedure TfServidores.VerificaMsgsEspera(ChaveContato: integer);
var Indice: integer;
    x: integer;
begin
    x := 0;
    while (x<DtModuleClient.MsgsEmEspera.Count) do
    begin
        if (DtModuleClient.ChavesMsgsEmEspera[x]=inttostr(ChaveContato)) then
        begin
            Indice := DtModuleClient.ChavesUsuarios.IndexOf(inttostr(ChaveContato));

DtModuleClient.Server.Socket.Connections[Indice].SendText(Protocolo(DtModule
Client.MsgsEmEspera[x]));
            DtModuleClient.ChavesMsgsEmEspera.Delete(x);
            DtModuleClient.MsgsEmEspera.Delete(x);
        end
        else
        begin

```

```

        inc(x);
    end;
end;
end;

// -----
//  Fim das outras funções
// -----

```

```
end.
```

```
unit ufsPrincipal;
```

```
interface
```

```
uses
```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Grids, ExtCtrls, TFlatPanelUnit, uDtModule, Menus, uFCadServidor,
    ComCtrls, uFCadAvatar, TFlatSpeedButtonUnit, StdCtrls, uFuncoes;

```

```
type
```

```

TfsPrincipal = class(TForm)
    Panel: TFlatPanel;
    Grid: TStringGrid;
    MainMenu1: TMainMenu;
    Ferramentas1: TMenuItem;
    CadastrarServidor1: TMenuItem;
    Sair1: TMenuItem;
    Sair2: TMenuItem;
    CarregarListaAvatares1: TMenuItem;

```

```

CarregarListaServidores1: TMenuItem;
CadastrarAvatar1: TMenuItem;
N1: TMenuItem;
StatusBar: TStatusBar;
AtivarServidor1: TMenuItem;
N2: TMenuItem;
btKick: TFlatSpeedButton;
btKickAll: TFlatSpeedButton;
btCadServidor: TFlatSpeedButton;
btCadAvatar: TFlatSpeedButton;
btSair: TFlatSpeedButton;
PopupMenu: TPopupMenu;
ManutenodoServidor1: TMenuItem;
Correodebuginesperado1: TMenuItem;
Servidorserreiniciado1: TMenuItem;
pnRight: TPanel;
procedure FormCreate(Sender: TObject);
procedure CadastrarServidor1Click(Sender: TObject);
procedure Sair2Click(Sender: TObject);
procedure CarregarListaServidores1Click(Sender: TObject);
procedure CarregarListaAvatares1Click(Sender: TObject);
procedure CadastrarAvatar1Click(Sender: TObject);
procedure btSairClick(Sender: TObject);
procedure btKickClick(Sender: TObject);
procedure btKickAllClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

var

fsPrincipal: TfsPrincipal;

implementation

{\$R *.dfm}

procedure TfsPrincipal.FormCreate(Sender: TObject);

begin

DtModule.PreencheListaAvatares();

DtModule.PreencheListaServidores();

Grid.Cells[0,0] := 'IP';

Grid.Cells[1,0] := 'E-mail';

Grid.Cells[2,0] := 'Conectado em';

Grid.Cells[3,0] := 'Tipo';

DtModule.Server.Active := true;

StatusBar.Panels.Items[0].Text := 'Status do Servidor: Ativo';

end;

procedure TfsPrincipal.CadastrarServidor1Click(Sender: TObject);

var form: TfcadServidor;

begin

form := TfcadServidor.Create(nil);

form.ShowModal;

freeandnil(form);

DtModule.PreencheListaServidores;

end;

procedure TfsPrincipal.Sair2Click(Sender: TObject);

begin

application.Terminate;

end;

```
procedure TfsPrincipal.CarregarListaServidores1Click(Sender: TObject);
begin
  DtModule.PreencheListaServidores;
end;
```

```
procedure TfsPrincipal.CarregarListaAvatares1Click(Sender: TObject);
begin
  DtModule.PreencheListaAvatares;
end;
```

```
procedure TfsPrincipal.CadastrarAvatar1Click(Sender: TObject);
var form: TfCadAvatar;
begin
  form := TfCadAvatar.Create(nil);
  form.ShowModal;
  freeandnil(form);
  DtModule.PreencheListaAvatares;
end;
```

```
procedure TfsPrincipal.btSairClick(Sender: TObject);
begin
  application.Terminate;
end;
```

```
procedure TfsPrincipal.btKickClick(Sender: TObject);
var p: pointer;
begin
  if (Grid.Row<>0) and (DtModule.Server.Socket.ActiveConnections>=Grid.Row)
  then
```

```

begin
  p := DtModule.Server.Socket.Connections[Grid.Row-1].Data;
  if (Application.MessageBox('    Tem certeza que deseja kickar este usuário?
', 'Kickar Usuário', 4)=idYES) then
    DtModule.KickarUsuario(DtModule.LocalizaCliente(p), "");
  end;
end;

```

```

procedure TfsPrincipal.btKickAllClick(Sender: TObject);
var x: integer;
begin
  if (Application.MessageBox('    Tem certeza que deseja kickar todos os
usuários?    ', 'Kickar Usuários', 4)=idYES) then
    for x := 0 to DtModule.Server.Socket.ActiveConnections-1 do
      DtModule.KickarUsuario(x, "");
    end;
end.

```

```

unit uFTP;

```

```

interface

```

```

uses graphics, controls, classes;

```

```

type

```

```

  TFTP=class(TComponent)

```

```

  private

```

```

    Arquivo: string;

```

```

    FFTPHost: string;

```

```

FFTPUser: string;
FFTPPass: string;
FFTPDir: string;
FFTPThumbDir: string;
FFTPImgDir: string;
FFTPFileDir: string;
procedure SetFTPHost(const Value: string);
procedure SetFTPUser(const Value: string);
procedure SetFTPPass(const Value: string);
procedure SetFTPDir(const Value: string);
procedure SetFTPFileDir(const Value: string);
procedure SetFTPImgDir(const Value: string);
procedure SetFTPThumbDir(const Value: string);
published
    property FTPHost: string read FFTPHost write SetFTPHost;
    property FTPUser: string read FFTPUser write SetFTPUser;
    property FTPPass: string read FFTPPass write SetFTPPass;
    property FTPDir: string read FFTPDir write SetFTPDir;
    property FTPImgDir: string read FFTPImgDir write SetFTPImgDir;
    property FTPThumbDir: string read FFTPThumbDir write SetFTPThumbDir;
    property FTPFileDir: string read FFTPFileDir write SetFTPFileDir;
public
    constructor Create(aOwner:tcomponent);override;
    destructor Destroy;override;

    procedure Carregar;
    procedure Gravar;
    procedure RestaurarPadrao;
end;

var

```


FTP: TFTP;

implementation

uses sysutils, uMensagemStatus, uFuncoes;

{ TConfiguracoes }

procedure TFTP.Carregar;

begin

 NovaMsgStatus('Carregando configurações...');

 try

 LerObjetoComoTexto(arquivo,self);

 finally

 RemoverMsgStatus;

 end;

end;

constructor TFTP.Create(aOwner:tcomponent);

begin

 inherited;

 restaurarPadrao;

end;

destructor TFTP.Destroy;

begin

 inherited;

end;

procedure TFTP.Gravar;

begin

```
    GravarObjetoComoTexto(arquivo,self);  
end;
```

```
procedure TFTP.RestaurarPadrao;  
begin  
    //valores padrão  
    FTPHost := 'ftp.colorgames.com.br';  
    FTPUser := 'colorgam';  
    FTPPass := '34col64';  
    FTPDir := 'www/3dblah/';  
    FTPImgDir := 'Users';  
    FTPThumbDir := 'Servers';  
    FTPFileDir := 'Files';  
end;
```

```
procedure TFTP.SetFTPSDir(const Value: string);  
begin  
    FFTPDir := Value;  
end;
```

```
procedure TFTP.SetFTPFileDir(const Value: string);  
begin  
    FFTPFileDir := Value;  
end;
```

```
procedure TFTP.SetFTPHost(const Value: string);  
begin  
    FFTPHost := Value;  
end;
```

```
procedure TFTP.SetFTPImgDir(const Value: string);
```

```
begin
  FFTPImgDir := Value;
end;

procedure TFTP.SetFTPPass(const Value: string);
begin
  FFTPPass := Value;
end;

procedure TFTP.SetFTPThumbDir(const Value: string);
begin
  FFTPThumbDir := Value;
end;

procedure TFTP.SetFTPUser(const Value: string);
begin
  FFTPUser := Value;
end;

initialization
  FTP := TFTP.create(nil);
  try
    FTP.arquivo:='FTP.cfg';
    FTP.carregar;
  except
  end;
finalization
  FTP.free;
end.
```

```
unit uFuncoes;
```

```
interface
```

```
uses Windows, Classes, uConstantes, ShellApi, jpeg, Graphics;
```

```
type
```

```
  TRGBArray = array[Word] of TRGBTriple;
```

```
  pRGBArray = ^TRGBArray;
```

```
  //extrai o codigo e a mensagem do protocolo
```

```
  procedure DecodificaString(const Msg: string; var Codigo: integer; var
    MsgDecodificada: string);
```

```
  //separa uma string em n listas baseado no caracter separador definido em
  uConstantes
```

```
  procedure SeparaString(Msg: string; var Vetor: TStringList);
```

```
  procedure SeparaString2(Msg: string; var Vetor1: TStringList; var Vetor2:
    TStringList);
```

```
  procedure SeparaString3(Msg: string; var Vetor1: TStringList; var Vetor2:
    TStringList; var Vetor3: TStringList);
```

```
  procedure SeparaString4(Msg: string; var Vetor1: TStringList; var Vetor2:
    TStringList; var Vetor3: TStringList; var Vetor4: TStringList);
```

```
  procedure SeparaString5(Msg: string; var Vetor1: TStringList; var Vetor2:
    TStringList; var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5:
    TStringList);
```

```
  procedure SeparaString6(Msg: string; var Vetor1: TStringList; var Vetor2:
    TStringList; var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5: TStringList;
    var Vetor6: TStringList);
```

```
procedure SeparaString7(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList; var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5: TStringList;
var Vetor6: TStringList; var Vetor7: TStringList);
```

```
//retorna a descricao do status
```

```
function Status(Indice: integer): string;
```

```
//formata a mensagem no protocolo usado
```

```
function Protocolo(Msg: string): string;
```

```
//Separa todas as mensagens de uma string
```

```
procedure DivideMsgs(Msg: string; var ListaMsgs: TStringList);
```

```
//executa um programa com parâmetros
```

```
procedure ExecuteProgram(Nome, Parametros: String);
```

```
//retorna a posicao Left de determinado frame
```

```
function FrameLeft(Frame: integer): integer;
```

```
//retorna a posicao Top de determinado frame
```

```
function FrameTop(Frame: integer): integer;
```

```
//força uma janela a ficar ativa
```

```
function ForceForegroundWindow(hwnd: THandle): Boolean;
```

```
//converte um Jpeg em Bmp
```

```
function JpgToBmp(iJpg: String; var iBmp: string): Boolean;
```

```
//converte um Bmp em Jpeg
```

```
function BmpToJpg(iBmp: String; var iJpg: string): Boolean;
```

```

//cria um Thumbnail de uma imagem Jpeg
procedure CreateThumbnail(InStream, OutStream: TStream; Width, Height:
Integer; FillColor: TColor=clWhite); overload;
procedure CreateThumbnail(const InFileName, OutFileName: string; Width,
Height: Integer; FillColor: TColor=clWhite); overload;

//substitui todas as ocorrencias do Caracter1 pelo Caracter2
function TrocaCaracter(var Frase: string; Caracter1: string; Caracter2: string):
string;

//redimensiona uma imagem
procedure SmoothResize(Src, Dst: TBitmap);

//carrega e salva uma imagem Jpeg
function LoadJPEGPictureFile(Bitmap: TBitmap; FilePath, FileName: string):
Boolean;
function SaveJPEGPictureFile(Bitmap: TBitmap; FilePath, FileName: string;
Quality: Integer): Boolean;

//redimensiona uma imagem
procedure ResizeImage(FileName: string; MaxWidth: Integer);

//retorna as dimensoes de uma imagem Jpeg
function JPEGDimensions(Filename : string; var X, Y : Word) : boolean;

//move um arquivo Jpeg para uma pasta de destino
procedure MoverJpeg(Origem: string; Destino: string);

//le e grava as propriedades de um objeto em um arquivo
procedure GravarObjetoComoTexto(arquivo:string;objeto:tcomponent);
function LerObjetoComoTexto(arquivo:string;objeto:tcomponent):tcomponent;

```

```
//retorna o tamanho do arquivo
function FileSize(Arquivo: string): integer;

//retorna todos os diretorios do Path
procedure GetDirList(Path: string; var Lista: TStringList);

//remove os caracteres passados no vetor da string Str
function RemoveCaracteres(Str: String; Caracteres: Array of String): String;
```

implementation

uses SysUtils;

```
procedure DecodificaString(const Msg: string; varCodigo: integer; var
MsgDecodificada: string);
begin
    Codigo := strtoint(copy(Msg,1,3));
    MsgDecodificada := copy(Msg,4,length(Msg)-3);
end;
```

```
procedure SeparaString(Msg: string; var Vetor: TStringList);
var x: integer;
    aux: string;
begin
    x := 1;
    repeat
        aux := '';
        while (Msg[x]<>separador) and (x<=length(Msg)) do
            begin
                aux := aux + Msg[x];
```

```

    inc(x);
end;
inc(x);
Vetor.Add(aux);
until (x>length(Msg));
if (Msg[x-1]=separador) then
    Vetor.Add("");
end;

```

```

procedure SeparaString2(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
    x := 1;
    v := 1;
    repeat
        aux := "";
        while (Msg[x]<>separador) and (x<=length(Msg)) do
            begin
                aux := aux + Msg[x];
                inc(x);
            end;
        inc(x);
        if (v=1) then
            begin
                Vetor1.Add(aux);
                inc(v);
            end
        else if (v=2) then

```



```

begin
  Vetor2.Add(aux);
  dec(v);
end;
until (x>length(Msg));
end;

```

```

procedure SeparaString3(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList; var Vetor3: TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
  x := 1;
  v := 1;
  repeat
    aux := "";
    while (Msg[x]<>separador) and (x<=length(Msg)) do
      begin
        aux := aux + Msg[x];
        inc(x);
      end;
    inc(x);
    if (v=1) then
      begin
        Vetor1.Add(aux);
        inc(v);
      end
    else if (v=2) then
      begin
        Vetor2.Add(aux);

```

```

    inc(v);
end
else if (v=3) then
begin
    Vetor3.Add(aux);
    v := 1;
end;
until (x>length(Msg));
end;

```

```

procedure SeparaString4(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList; var Vetor3: TStringList; var Vetor4: TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
    x := 1;
    v := 1;
    repeat
        aux := "";
        while (Msg[x]<>separador) and (x<=length(Msg)) do
            begin
                aux := aux + Msg[x];
                inc(x);
            end;
        inc(x);
        if (v=1) then
            begin
                Vetor1.Add(aux);
                inc(v);
            end

```

```

else if (v=2) then
begin
  Vetor2.Add(aux);
  inc(v);
end
else if (v=3) then
begin
  Vetor3.Add(aux);
  inc(v);
end
else if (v=4) then
begin
  Vetor4.Add(aux);
  v := 1;
end;
until (x>length(Msg));
end;

```

```

procedure SeparaString5(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList;
  var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5: TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
  x := 1;
  v := 1;
  repeat
    aux := "";
    while (Msg[x]<>separador) and (x<=length(Msg)) do
      begin

```

```
    aux := aux + Msg[x];
    inc(x);
end;
inc(x);
if (v=1) then
begin
    Vetor1.Add(aux);
    inc(v);
end
else if (v=2) then
begin
    Vetor2.Add(aux);
    inc(v);
end
else if (v=3) then
begin
    Vetor3.Add(aux);
    inc(v);
end
else if (v=4) then
begin
    Vetor4.Add(aux);
    inc(v);
end
else if (v=5) then
begin
    Vetor5.Add(aux);
    v := 1;
end;
until (x>length(Msg));
end;
```

```

procedure SeparaString6(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList;
    var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5: TStringList;
    var Vetor6: TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
    x := 1;
    v := 1;
    repeat
        aux := '';
        while (Msg[x]<>separador) and (x<=length(Msg)) do
            begin
                aux := aux + Msg[x];
                inc(x);
            end;
            inc(x);
            if (v=1) then
                begin
                    Vetor1.Add(aux);
                    inc(v);
                end
            else if (v=2) then
                begin
                    Vetor2.Add(aux);
                    inc(v);
                end
            else if (v=3) then
                begin

```

```

    Vetor3.Add(aux);
    inc(v);
end
else if (v=4) then
begin
    Vetor4.Add(aux);
    inc(v);
end
else if (v=5) then
begin
    Vetor5.Add(aux);
    inc(v);
end
else if (v=6) then
begin
    Vetor6.Add(aux);
    v := 1;
end;
until (x>length(Msg));
end;

```

```

procedure SeparaString7(Msg: string; var Vetor1: TStringList; var Vetor2:
TStringList;
    var Vetor3: TStringList; var Vetor4: TStringList; var Vetor5: TStringList;
    var Vetor6: TStringList; var Vetor7: TStringList);
var x: integer;
    aux: string;
    v: integer;
begin
    x := 1;
    v := 1;

```

```
repeat
  aux := "";
  while (Msg[x]<>separador) and (x<=length(Msg)) do
  begin
    aux := aux + Msg[x];
    inc(x);
  end;
  inc(x);
  if (v=1) then
  begin
    Vetor1.Add(aux);
    inc(v);
  end
  else if (v=2) then
  begin
    Vetor2.Add(aux);
    inc(v);
  end
  else if (v=3) then
  begin
    Vetor3.Add(aux);
    inc(v);
  end
  else if (v=4) then
  begin
    Vetor4.Add(aux);
    inc(v);
  end
  else if (v=5) then
  begin
    Vetor5.Add(aux);
```

```

    inc(v);
end
else if (v=6) then
begin
    Vetor6.Add(aux);
    inc(v);
end
else if (v=7) then
begin
    Vetor7.Add(aux);
    v := 1;
end;
until (x>length(Msg));
end;

```

```

function Status(Indice: integer): string;
begin
    if (Indice=-1) then
        result := 'Offline'
    else if (Indice=0) then
        result := 'Ao Telefone'
    else if (Indice=1) then
        result := 'Ausente'
    else if (Indice=2) then
        result := 'Comendo'
    else if (Indice=3) then
        result := 'Ocupado'
    else if (Indice=4) then
        result := 'Online';
end;

```



```
function Protocolo(Msg: string): string;
```

```
var tam: integer;
```

```
    tamanho: string;
```

```
begin
```

```
    tam := length(Msg);
```

```
    tamanho := inttostr(tam);
```

```
    result := tamanho+SeparadorTamanho+Msg;
```

```
end;
```

```
procedure DivideMsgs(Msg: string; var ListaMsgs: TStringList);
```

```
var x: integer;
```

```
    tamanho: integer;
```

```
    PosAnt: integer;
```

```
begin
```

```
    PosAnt := 1;
```

```
    while PosAnt<length(Msg) do
```

```
    begin
```

```
        for x := PosAnt to length(Msg) do
```

```
        begin
```

```
            if (Msg[x]='ç') then
```

```
                break;
```

```
        end;
```

```
        tamanho := strtoint(copy(Msg, PosAnt, x-PosAnt));
```

```
        PosAnt := x+tamanho+1;
```

```
        ListaMsgs.Add(copy(Msg, x+1, tamanho));
```

```
    end;
```

```
end;
```

```
procedure ExecuteProgram(Nome, Parametros: String);
```

```
var Comando : array[0..1024] of Char;
```

```
    Pargs : array[0..1024] of Char;
```

```

begin
  StrPCopy(Comando, Nome);
  StrPCopy(Parms, Parametros);
  ShellExecute(0, nil, Comando, Parms, nil, sw_shownormal);
end;

```

```

function FrameLeft(Frame: integer): integer;
begin
  if ((Frame mod 2)=1) then
    result := 16
  else
    result := 370;
end;

```

```

function FrameTop(Frame: integer): integer;
begin
  if (Frame=1) or (Frame=2) then
    result := 20
  else if (Frame=3) or (Frame=4) then
    result := 146
  else if (Frame=5) or (Frame=6) then
    result := 272
  else if (Frame=7) or (Frame=8) then
    result := 398;
end;

```

```

function ForceForegroundWindow(hwnd: THandle): Boolean;
const
  SPI_GETFOREGROUNDLOCKTIMEOUT = $2000;
  SPI_SETFOREGROUNDLOCKTIMEOUT = $2001;
var

```

```

ForegroundThreadID: DWORD;
ThisThreadID: DWORD;
timeout: DWORD;
begin
  if (IsIconic(hwnd)) then
    ShowWindow(hwnd, SW_RESTORE);
  if (GetForegroundWindow = hwnd) then
    Result := True
  else
    begin
      // Windows 98/2000 doesn't want to foreground a window when some other
      // window has keyboard focus
      if ((Win32Platform = VER_PLATFORM_WIN32_NT) and (Win32MajorVersion >
4)) or
        ((Win32Platform = VER_PLATFORM_WIN32_WINDOWS) and
        ((Win32MajorVersion > 4) or ((Win32MajorVersion = 4) and
        (Win32MinorVersion > 0)))) then
        begin
          // Code from Karl E. Peterson, www.mvps.org/vb/sample.htm
          // Converted to Delphi by Ray Lischner
          // Published in The Delphi Magazine 55, page 16
          Result := False;
          ForegroundThreadID := GetWindowThreadProcessID(GetForegroundWindow,
nil);
          ThisThreadID := GetWindowThreadPRocessId(hwnd, nil);
          if AttachThreadInput(ThisThreadID, ForegroundThreadID, True) then
            begin
              BringWindowToTop(hwnd); // IE 5.5 related hack
              SetForegroundWindow(hwnd);
              AttachThreadInput(ThisThreadID, ForegroundThreadID, False);
              Result := (GetForegroundWindow = hwnd);
            end
          end
        end
      end
    end
  end
end

```

```

end;
if not Result then
begin
  // Code by Daniel P. Stasinski
  SystemParametersInfo(SPI_GETFOREGROUNDLOCKTIMEOUT, 0,
@timeout, 0);
  SystemParametersInfo(SPI_SETFOREGROUNDLOCKTIMEOUT, 0,
TObject(0),
  SPIF_SENDCHANGE);
  BringWindowToTop(hwnd); // IE 5.5 related hack
  SetForegroundWindow(hWnd);
  SystemParametersInfo(SPI_SETFOREGROUNDLOCKTIMEOUT, 0,
TObject(timeout), SPIF_SENDCHANGE);
end;
end
else
begin
  BringWindowToTop(hwnd); // IE 5.5 related hack
  SetForegroundWindow(hwnd);
end;
Result := (GetForegroundWindow = hwnd);
end;
end;

```

```

function JpgToBmp(iJpg: String; var iBmp: string): Boolean;
var MyJPEG: TJPEGImage;
    MyBMP: TBitmap;
begin
  Result := False;
  if fileExists(iJpg) then
  begin

```

```

MyJPEG := TJPEGImage.Create;
try
  MyJpeg.LoadFromFile(iJpg);
  MyBMP := TBitmap.Create;
  with MyBMP do
    begin
      Width := MyJPEG.Width;
      Height := MyJPEG.Height;
      Canvas.Draw(0,0,MyJPEG);
      iBmp := copy(iJpg,1,length(iJpg)-length(extractfileext(iJpg))+1)+'.Bmp';
      SaveToFile(iBmp);
      Free;
      Result := True;
    end;
  finally
    MyJpeg.Free;
  end;
end;
end;

```

```

function BmpToJpg(iBmp: String; var iJpg: string):Boolean;
var MyJPEG : TJPEGImage;
    MyBMP : TBitmap;
begin
  Result := False;
  if fileExists(iBmp) then
    begin
      MyBMP := TBitmap.Create;
      with MyBMP do
        begin
          try

```

```

    LoadFromFile(iBmp);
    MyJPEG := TJPEGImage.Create;
    with MyJPEG do
    begin
        assign(MyBMP);
        iJpg := copy(iBmp,1,length(iBmp)-4)+'.jpeg';
        SaveToFile(iJpg);
        Free;
        Result := True;
    end;
finally
    Free;
end;
end;
end;
end;

procedure CreateThumbnail(InStream, OutStream: TStream;
    Width, Height: Integer; FillColor: TColor=clWhite);
var
    JpegImage: TJpegImage;
    Bitmap: TBitmap;
    Ratio: Double;
    ARect: TRect;
    AHeight, AHeightOffset: Integer;
    AWidth, AWidthOffset: Integer;
begin
    // Check for invalid parameters
    if Width<1 then
        raise Exception.Create('Invalid Width');
    if Height<1 then

```

```

    raise Exception.Create('Invalid Height');
    JpegImage:=TJpegImage.Create;
    try
// Load the image
    JpegImage.LoadFromStream(InStream);
// Create bitmap, and calculate parameters
    Bitmap:=TBitmap.Create;
    try
        Ratio:=JpegImage.Width/JpegImage.Height;
        if Ratio>1 then
            begin
                AHeight:=Round(Width/Ratio);
                AHeightOffset:=(Height-AHeight) div 2;
                AWidth:=Width;
                AWidthOffset:=0;
            end
        else
            begin
                AWidth:=Round(Height*Ratio);
                AWidthOffset:=(Width-AWidth) div 2;
                AHeight:=Height;
                AHeightOffset:=0;
            end;
        Bitmap.Width:=Width;
        Bitmap.Height:=Height;
        Bitmap.Canvas.Brush.Color:=FillColor;
        Bitmap.Canvas.FillRect(Rect(0,0,Width,Height));
// StretchDraw original image

    ARect:=Rect(AWidthOffset,AHeightOffset,AWidth+AWidthOffset,AHeight+AHeight
Offset);

```

```

        Bitmap.Canvas.StretchDraw(ARect,JpegImage);
// Assign back to the Jpeg, and save to the file
        JpegImage.Assign(Bitmap);
        JpegImage.SaveToStream(OutStream);
    finally
        Bitmap.Free;
    end;
finally
    JpegImage.Free;
end;
end;

procedure CreateThumbnail(const InFileName, OutFileName: string;
    Width, Height: Integer; FillColor: TColor=clWhite); overload;
var
    InStream, OutStream: TFileStream;
begin
    InStream:=TFileStream.Create(InFileName,fmOpenRead);
    try
        OutStream:=TFileStream.Create(OutFileName,fmOpenWrite or fmCreate);
        try
            CreateThumbnail(InStream,OutStream,Width,Height,FillColor);
        finally
            OutStream.Free;
        end;
    finally
        InStream.Free;
    end;
end;

```



```

function TrocaCaracter(var Frase: string; Caracter1: string; Caracter2: string):
string;
var x: integer;
    aux: string;
begin
    aux := "";
    for x := 1 to length(Frase) do
    begin
        if (copy(Frase,x,1)=Caracter1) then
            aux := aux+Caracter2
        else
            aux := aux+copy(Frase,x,1);
        end;
    result := aux;
end;

```

```

procedure SmoothResize(Src, Dst: TBitmap);
var
    x, y: Integer;
    xP, yP: Integer;
    xP2, yP2: Integer;
    SrcLine1, SrcLine2: pRGBArray;
    t3: Integer;
    z, z2, iz2: Integer;
    DstLine: pRGBArray;
    DstGap: Integer;
    w1, w2, w3, w4: Integer;
begin
    Src.PixelFormat := pf24Bit;
    Dst.PixelFormat := pf24Bit;

```

```

if (Src.Width = Dst.Width) and (Src.Height = Dst.Height) then
  Dst.Assign(Src)
else
begin
  DstLine := Dst.ScanLine[0];
  DstGap := Integer(Dst.ScanLine[1]) - Integer(DstLine);

  xP2 := MulDiv(pred(Src.Width), $10000, Dst.Width);
  yP2 := MulDiv(pred(Src.Height), $10000, Dst.Height);
  yP := 0;

  for y := 0 to pred(Dst.Height) do
  begin
    xP := 0;

    SrcLine1 := Src.ScanLine[yP shr 16];

    if (yP shr 16 < pred(Src.Height)) then
      SrcLine2 := Src.ScanLine[succ(yP shr 16)]
    else
      SrcLine2 := Src.ScanLine[yP shr 16];

    z2 := succ(yP and $FFFF);
    iz2 := succ((not yp) and $FFFF);
    for x := 0 to pred(Dst.Width) do
    begin
      t3 := xP shr 16;
      z := xP and $FFFF;
      w2 := MulDiv(z, iz2, $10000);
      w1 := iz2 - w2;
      w4 := MulDiv(z, z2, $10000);

```

```

w3 := z2 - w4;
DstLine[x].rgbtRed := (SrcLine1[t3].rgbtRed * w1 +
  SrcLine1[t3 + 1].rgbtRed * w2 +
  SrcLine2[t3].rgbtRed * w3 + SrcLine2[t3 + 1].rgbtRed * w4) shr 16;
DstLine[x].rgbtGreen :=
  (SrcLine1[t3].rgbtGreen * w1 + SrcLine1[t3 + 1].rgbtGreen * w2 +
  SrcLine2[t3].rgbtGreen * w3 + SrcLine2[t3 + 1].rgbtGreen * w4) shr 16;
DstLine[x].rgbtBlue := (SrcLine1[t3].rgbtBlue * w1 +
  SrcLine1[t3 + 1].rgbtBlue * w2 +
  SrcLine2[t3].rgbtBlue * w3 +
  SrcLine2[t3 + 1].rgbtBlue * w4) shr 16;
Inc(xP, xP2);
end;
Inc(yP, yP2);
DstLine := pRGBArray(Integer(DstLine) + DstGap);
end;
end;
end;

function LoadJPEGPictureFile(Bitmap: TBitmap; FilePath, FileName: string):
Boolean;
var
  JPEGImage: TJPEGImage;
begin
  if (FileName = "") then // No FileName so nothing
    Result := False //to load - return False...
  else
    begin
      try // Start of try except
        JPEGImage := TJPEGImage.Create; // Create the JPEG image... try // now

```

```

try // to load the file but
    JPEGImage.LoadFromFile(FilePath + FileName);
    // might fail...with an Exception.
    Bitmap.Assign(JPEGImage);
    // Assign the image to our bitmap.Result := True;
    // Got it so return True.
finally
    JPEGImage.Free; // ...must get rid of the JPEG image. finally
end; {try}
except
    Result := False; // Oops...never Loaded, so return False.
end;
end;
end;

function SaveJPEGPictureFile(Bitmap: TBitmap; FilePath, FileName: string;
    Quality: Integer): Boolean;
begin
    Result := True;
    try
        if ForceDirectories(FilePath) then
            begin
                with TJPEGImage.Create do
                    begin
                        try
                            Assign(Bitmap);
                            CompressionQuality := Quality;
                            SaveToFile(FilePath + FileName);
                        finally
                            Free;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        end;
    end;
except
    raise;
    Result := False;
end;
end;

procedure ResizeImage(FileName: string; MaxWidth: Integer);
var
    OldBitmap: TBitmap;
    NewBitmap: TBitmap;
    aWidth: Integer;
begin
    OldBitmap := TBitmap.Create;
    try
        if LoadJPEGPictureFile(OldBitmap, ExtractFilePath(FileName),
            ExtractFileName(FileName)) then
            begin
                aWidth := OldBitmap.Width;
                if (OldBitmap.Width > MaxWidth) then
                    begin
                        aWidth := MaxWidth;
                        NewBitmap := TBitmap.Create;
                        try
                            NewBitmap.Width := MaxWidth;
                            NewBitmap.Height := MulDiv(MaxWidth, OldBitmap.Height,
                                OldBitmap.Width);
                            SmoothResize(OldBitmap, NewBitmap);
                            RenameFile(FileName, ChangeFileExt(FileName, '.$$$'));
                            if SaveJPEGPictureFile(NewBitmap, ExtractFilePath(FileName),

```

```

        ExtractFileName(FileName), 75) then
        DeleteFile(ChangeFileExt(FileName, '.$$$'))
    else
        RenameFile(ChangeFileExt(FileName, '.$$$'), FileName);
    finally
        NewBitmap.Free;
    end;
end;
end;
finally
    OldBitmap.Free;
end;
end;

function JPEGDimensions(Filename : string; var X, Y : Word) : boolean;
var
    SegmentPos : Integer;
    SOIcount : Integer;
    b : byte;
begin
    Result := False;
    with TFileStream.Create(Filename, fmOpenRead or fmShareDenyNone) do
    begin
        try
            Position := 0;
            Read(X, 2);
            if (X <> $D8FF) then
                exit;
            SOIcount := 0;
            Position := 0;
            while (Position + 7 < Size) do

```

```

begin
  Read(b, 1);
  if (b = $FF) then begin
    Read(b, 1);
    if (b = $D8) then
      inc(SOLcount);
    if (b = $DA) then
      break;
  end;
end;
if (b <> $DA) then
  exit;
SegmentPos := -1;
Position := 0;
while (Position + 7 < Size) do
begin
  Read(b, 1);
  if (b = $FF) then
  begin
    Read(b, 1);
    if (b in [$C0, $C1, $C2]) then
    begin
      SegmentPos := Position;
      dec(SOLcount);
      if (SOLcount = 0) then
        break;
    end;
  end;
end;
if (SegmentPos = -1) then
  exit;

```

```

    if (Position + 7 > Size) then
        exit;
    Position := SegmentPos + 3;
    Read(Y, 2);
    Read(X, 2);
    X := Swap(X);
    Y := Swap(Y);
    Result := true;
finally
    Free;
end;
end;
end;

```

```

procedure MoverJpeg(Origem: string; Destino: string);
var Jpeg: TJpegImage;
begin
    DeleteFile(Destino);
    Jpeg := TJpegImage.Create;
    Jpeg.LoadFromFile(Origem);
    Jpeg.SaveToFile(Destino);
    freeandnil(Jpeg);
    DeleteFile(Origem);
end;

```

```

{**

```

Grava um objeto TComponent num arquivo, no formato texto,
 Como os arquivos .dfm do delphi 5.
 Útil para guardar/carregar alguns parâmetros de
 configuração a partir de arquivos, basta criar uma
 classe em que as informações a serem gravadas são

propriedades published. E utilizar

<tt>GravarObjetoComoTexto()</tt> e <tt>LerObjetoComoTexto()</tt>

para gravar/carregar as informações.<p>

EPH-jan-mar2001

@param arquivo O nome do arquivo

@param objeto O objeto a ser armazenado

@see LerObjetoComoTexto

}

procedure GravarObjetoComoTexto(arquivo:string;objeto:tcomponent);

var

 binario:tstream;

 texto:tstream;

begin

 binario:=nil;

 texto:=nil;

 try

 binario:=tmemorystream.create;

 texto:=tfilestream.create(arquivo,fmCreate);

 binario.writecomponent(objeto);

 binario.Position:=0;

 ObjectBinaryToText(binario,texto);//grava como texto

 finally

 binario.free;

 texto.free;

 end;

end;

{**

Carrega um objeto que foi armazenado com

<tt>GravarObjetoComoTexto()</tt> de um arquivo.<p>

EPH-jan-mar2001

```

@param arquivo Nome do arquivo
@param objeto Objeto a ser carregado
@see GravarObjetoComoTexto
}
function LerObjetoComoTexto(arquivo:string;objeto:tcomponent):tcomponent;
var
  binario:tstream;
  texto:tstream;
begin
  if (not FileExists(arquivo)) then
    exit;
  binario:=nil;
  texto:=nil;
  try
    texto:=tfilestream.create(arquivo,fmOpenRead);
    binario:=tmemorystream.create;
    ObjectTextToBinary(texto,binario);
    binario.position:=0;
    result:=binario.ReadComponent(objeto);
  finally
    binario.free;
    texto.free;
  end;
end;

function FileSize(Arquivo: string): integer;
var ArqTmp : TMemoryStream;
begin
  result := -1;

```

```

if (FileExists(Arquivo)) then
begin
  ArqTmp := TMemoryStream.Create;
  ArqTmp.Clear;
  ArqTmp.LoadFromFile(Arquivo);
  result := ArqTmp.Size;
  freeandnil(ArqTmp);
end;
end;

```

```

procedure GetDirList(Path: string; var Lista: TStringList);
var x: integer;
    aux: string;
begin
  aux := "";
  for x := 1 to length(Path) do
  begin
    if (copy(Path, x, 1)='/') then
    begin
      Lista.Add(aux);
      aux := "";
    end
    else
      aux := aux + copy(Path, x, 1);
    end;
  if (aux<>") then
    Lista.Add(aux);
  end;
end;

```

```

function RemoveCaracteres(Str: String; Caracteres: Array of String): String;
var

```

```

    I: Integer;
begin
    Result := Str;
    for I := 0 to High(Caracteres) do
    begin
        while Pos(Caracteres[I], Result) > 0 do
            Delete(Result, Pos(Caracteres[I], Result), 1);
        end;
    end;
end;

end.

unit uJpeg;

interface

uses classes, windows, graphics, jpeg;

procedure
SaveBMPasJPEG(bm:tbitmap;dest:tstream;qualidade:TJPEGQualityRange=90);

procedure
SaveAsJpegToFile(bm:tbitmap;arq:string;qualidade:TJPEGQualityRange=90);

procedure LoadJPEGasBMP(src:tstream;bm:tbitmap);

procedure LoadJPEGFromFile(arq:string;bm:tbitmap);

procedure ConvertBMPToJPG (ArqOrigem: string; ArqDestino: string);

```

```
procedure ConvertICOtoBMP(ArqOrigem: string; ArqDestino: string);
```

```
procedure CreateThumbnail(InStream, OutStream: TStream;
  Width, Height: Integer; FillColor: TColor=clWhite); overload;
```

```
procedure CreateThumbnail(const InFileName, OutFileName: string;
  Width, Height: Integer; FillColor: TColor=clWhite); overload;
```

```
implementation
```

```
uses sysutils;
```

```
procedure
```

```
SaveBMPasJPEG(bm:tbitmap;dest:tstream;qualidade:TJPEGQualityRange);
```

```
begin
```

```
  with tjpegimage.create do
```

```
    try
```

```
      CompressionQuality:=qualidade;
```

```
      assign(bm);
```

```
      compress;
```

```
      savetostream(dest);
```

```
    finally
```

```
      free;
```

```
    end;
```

```
end;
```

```
procedure LoadJPEGasBMP(src:tstream;bm:tbitmap);
```

```
var
```

```
  ji:tjpegimage;
```

```
begin
```

```
  ji:=tjpegimage.create;
```

```

try
  ji.loadfromstream(src);
  bm.assign(ji);
finally

  ji.free;
end;
end;

procedure LoadJPEGFromFile(arq:string;bm:tbitmap);
var
  st:tfilestream;
begin
  st:=tfilestream.create(arq,fmOpenRead);
  try
    loadjpegAsbmp(st,bm);
  finally
    st.free;
  end;
end;

procedure
SaveAsJpegToFile(bm:tbitmap;arq:string;qualidade:TJPEGQualityRange=90);
var
  st:tstream;
begin
  st:=tfilestream.create(arq,fmCreate);
  try
    SaveBMPasJPEG(bm,st,qualidade);
  finally
    st.free;
  end;
end;

```

```

    end;
end;

procedure ConvertBMPToJPG (ArqOrigem: string; ArqDestino: string);
var BMP: tbitmap; Jpeg: tJpegImage;
begin
    BMP := TBitmap.Create;
    JPEG := TJPEGImage.Create;
    try
        BMP.LoadFromFile(ArqOrigem);
        JPEG.Assign(BMP);
        Jpeg.SaveToFile(ArqDestino);
    finally
        freeandnil(BMP);
        freeandnil(JPEG);
    end;
end;

procedure ConvertICOtoBMP(ArqOrigem: string; ArqDestino: string);
var
    BitMap : TBitmap;
    Icon : TIcon;
begin
    Bitmap := TBitmap.Create;
    Icon := TIcon.Create;
    TRY
        // lê arquivo ICO
        icoN.LoadFromFile(ArqOrigem);
        with bitmap do begin
            // seta tamanho do bitmap igual ao do ícone
            Bitmap.Width := Icon.Width;

```

```

    Bitmap.Height := Icon.Height;
// desenha no canvas do bitmap
    Bitmap.Canvas.Draw(0,0,Icon);
// Salva BMP
    SaveToFile(ArqDestino);
    end;
finally
    BitMap.Free;
    icoN.Free;
    end;
end;

procedure CreateThumbnail(InStream, OutStream: TStream;
    Width, Height: Integer; FillColor: TColor=clWhite);
var
    JpegImage: TJpegImage;
    Bitmap: TBitmap;
    Ratio: Double;
    ARect: TRect;
    AHeight, AHeightOffset: Integer;
    AWidth, AWidthOffset: Integer;
begin
// Check for invalid parameters
    if Width<1 then
        raise Exception.Create('Invalid Width');
    if Height<1 then
        raise Exception.Create('Invalid Height');
    JpegImage:=TJpegImage.Create;
    try
// Load the image
        JpegImage.LoadFromStream(InStream);

```



```

// Create bitmap, and calculate parameters
Bitmap:=TBitmap.Create;
try
  Ratio:=JpegImage.Width/JpegImage.Height;
  if Ratio>1 then
    begin
      AHeight:=Round(Width/Ratio);
      AHeightOffset:=(Height-AHeight) div 2;
      AWidth:=Width;
      AWidthOffset:=0;
    end
  else
    begin
      AWidth:=Round(Height*Ratio);
      AWidthOffset:=(Width-AWidth) div 2;
      AHeight:=Height;
      AHeightOffset:=0;
    end;
  Bitmap.Width:=Width;
  Bitmap.Height:=Height;
  Bitmap.Canvas.Brush.Color:=FillColor;
  Bitmap.Canvas.FillRect(Rect(0,0,Width,Height));
// StretchDraw original image

ARect:=Rect(AWidthOffset,AHeightOffset,AWidth+AWidthOffset,AHeight+AHeight
Offset);
  Bitmap.Canvas.StretchDraw(ARect,JpegImage);
// Assign back to the Jpeg, and save to the file
  JpegImage.Assign(Bitmap);
  JpegImage.SaveToStream(OutStream);
finally

```

```

        Bitmap.Free;
    end;
finally
    JpegImage.Free;
end;
end;

procedure CreateThumbnail(const InFileName, OutFileName: string;
    Width, Height: Integer; FillColor: TColor=clWhite); overload;
var
    InStream, OutStream: TFileStream;
begin
    InStream:=TFileStream.Create(InFileName,fmOpenRead);
    try
        OutStream:=TFileStream.Create(OutFileName,fmOpenWrite or fmCreate);
        try
            CreateThumbnail(InStream,OutStream,Width,Height,FillColor);
        finally
            OutStream.Free;
        end;
    finally
        InStream.Free;
    end;
end;

end.

unit uMapa;

interface

```

```
uses SysUtils, Classes;
```

```
type
```

```
  TMapa = class(TObject)
```

```
    ChaveServidor: integer;
```

```
    Titulo: string;
```

```
    Descricao: string;
```

```
    NumUsuarios: integer;
```

```
    NumMaxUsuarios: integer;
```

```
    HostName: string;
```

```
    NomeArquivoMapa: string;
```

```
    NomeArquivoTextura: string;
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    procedure PreencheClasse(Lista: TStringList);
```

```
    { Public declarations }
```

```
end;
```

```
implementation
```

```
{ TMapa }
```

```
procedure TMapa.PreencheClasse(Lista: TStringList);
```

```
begin
```

```
  ChaveServidor := strtoint(Lista[0]);
```

```
  Titulo := Lista[1];
```

```
  Descricao := Lista[2];
```

```
  NumUsuarios := 0;
```

```
  NumMaxUsuarios := strtoint(Lista[3]);
```

```
    HostName := Lista[4];  
    NomeArquivoMapa := Lista[5];  
    NomeArquivoTextura := Lista[6];  
end;
```

```
end.
```

```
unit uModelo;
```

```
interface
```

```
type
```

```
    TModelo = class(TObject)  
        chave_modelo: integer;  
        nome_arquivo: string;  
        num_animacao: integer;  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;
```

```
implementation
```

```
end.
```

```
unit uUsuario;
```

```
interface
```

uses Classes, SysUtils, uModelo, Graphics, uBaixaImagemUsuario, ExtCtrls, Dialogs;

type

TUsuario = class(TObject)

 chave_usuario: integer;

 email: string;

 nome: string;

 apelido: string;

 nascimento: tdatetime;

 sexo: integer;

 estado_civil: integer;

 cidade: string;

 estado: string;

 pais: integer;

 telefone: string;

 celular: string;

 status: integer;

 foto: TPicture;

 modelo: TModelo;

 posicao: integer;

 constructor Create();

 destructor Destroy(); override;

private

 { Private declarations }

public

 Th: TBaixaImagemUsuario;

 procedure PreencheClasse(Lista: TStringList);

 procedure CarregarImagem(Image: TImage);

 { Public declarations }

end;

implementation

```
constructor TUsuario.Create();  
begin  
    inherited;  
    Foto := TPicture.Create;  
    Modelo := TModelo.Create;  
end;
```

```
destructor TUsuario.Destroy();  
begin  
    inherited;  
    if (Th<>nil) then  
        Th.Terminate;  
    freeandnil(Foto);  
    freeandnil(Modelo);  
end;
```

```
procedure TUsuario.PreencheClasse(Lista: TStringList);  
begin  
    chave_usuario := strtoint(Lista.Strings[0]);  
    nome := Lista.Strings[1];  
    apelido := Lista.Strings[2];  
    nascimento := strtodate(Lista.Strings[3]);  
    sexo := strtoint(Lista.Strings[4]);  
    estado_civil := strtoint(Lista.Strings[5]);  
    cidade := Lista.Strings[6];  
    estado := Lista.Strings[7];  
    pais := strtoint(Lista.Strings[8]);  
    telefone := Lista.Strings[9];
```

```

    celular := Lista.Strings[10];
    status := strtoint(Lista.Strings[11]);
    modelo.chave_modelo := strtoint(Lista.Strings[12]);
    modelo.nome_arquivo := Lista.Strings[13];
    email := Lista.Strings[14];
end;

procedure TUsuario.CarregarImagem(Image: TImage);
begin
    Th := TBaixaImagemUsuario.Criar(Foto, Image);
    Th.ChaveUsuario := Chave_Usuario;
    Th.Resume;
end;

end.

unit uUsuarioConectado;

interface

uses SysUtils, TrueVision3D, uModelo, Forms;

type
    TUsuarioConectado = class(TObject)
        Chave_Usuario: integer;
        Apelido: string;
        Modelo: TModelo;
        Personagem: TTvActor2;
        constructor Create();
        destructor Destroy(); override;
    end;

```

```

private
    { Private declarations }
public
    procedure LoadAvatar();
    { Public declarations }
end;

implementation

{ TUsuarioConectado }

constructor TUsuarioConectado.Create;
begin
    inherited;
    Modelo := TModelo.Create;
    Personagem := TTvActor2.Create(nil);
end;

destructor TUsuarioConectado.Destroy;
begin
    inherited;
    freeandnil(Modelo);
    freeandnil(Personagem);
end;

procedure TUsuarioConectado.LoadAvatar;
var Arquivo: string;
begin
    Arquivo :=
extractfilepath(application.ExeName)+'Models\' + Modelo.nome_arquivo + '\Model.md
l';

```



```
if (FileExists(Arquivo)) then
    Personagem.Load(Arquivo,Apelido,false,true)
else

Personagem.Load(extractfilepath(application.ExeName)+'Models\Default\Model.m
dl',Apelido,false,true);
    Personagem.SetAnimationByName('idle1');
    Personagem.PlayAnimation(10);
    Personagem.SetPosition(0, 0, 0);
end;

end.
```

ANEXO VI
CODIFICAÇÃO DO SITE

Index.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>| 3D BLAH |</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bottommargin="0" leftmargin="0" topmargin="0" rightmargin="0">
<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
<td align="center" valign="top" background="imagens/fundo.gif">
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td height="25">&nbsp;</td>
</tr>
</table>
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0" width="760" height="128">
<param name="movie" value="topo.swf">
<param name="quality" value="high">
<embed src="topo.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" width="760" height="128"></embed></object></td>
</tr>
</table>
<?php
```

```

//Inclui uma mensagem se o usuário estiver logado
error_reporting(0);
session_start();
if(isset($_SESSION['idblah'])){
    $nome = $_SESSION['nomeblah'];
    print("<table width=\"760\" border=\"0\" cellpadding=\"7\" background=\"imagens/corpo.gif\">
<tr>
    <td height=\"15\" valign=\"top\">
        <table width=\"100%\" border=\"0\" cellpadding=\"2\">
            <tr>
                <td bgcolor=\"#eeeeee\"
align=\"right\"><font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>$nome</b>, você está logado. <a href=\"logout.php\" style=\"color:
#000000\"><b>Sair</b></a>&nbsp;&nbsp;&nbsp;</font></td>
            </tr>
        </table>
    </td>
</tr>
</table>");
}
?>
<table width="760" border="0" cellpadding="0">
<tr>
<td height="270" align="left" valign="top" background="imagens/corpo.gif">
    <?php
        //Rotina para incluir uma página que é passado pela
variavel pagina
        $x = isset($_GET["pagina"]) ? $_GET["pagina"].".php" :
"inicial.php";
        file_exists($x) ? include($x) : include("erro404.php");
    </td>
</tr>
</table>

```

```

        ?>
    </td>
</tr>
</table>
<table width="760" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0" width="760" height="100">
            <param name="movie" value="rodape.swf">
            <param name="quality" value="high">
            <embed src="rodape.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" width="760" height="100"></embed></object></td>
        </tr>
    </table></td>
</tr>
</table>
</body>
</html>

```

Inicial.php

```

<table width="760" border="0" cellspacing="20" cellpadding="0">
    <tr>
        <td>
            <table width="100%" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td width="450" valign="top"><font size="4" face="Verdana, Arial, Helvetica,
sans-serif"><strong>O

```

```

</strong><font color="#FF9900" size="6"><strong>3D
BLAH</strong></font><strong>
veio para revolucionar<br>
o sistema de mensagens instant&acirc;neas<br><br></strong></font></td>
<td width="290" valign="top"><font size="4" face="Verdana, Arial, Helvetica,
sans-serif"><strong>Not&iacute;cias
<font color="#FF9900">3D BLAH</font></strong></font><br>
<textarea name="textarea" cols="40" rows="15" readonly wrap="VIRTUAL"
style="font-face: Verdana; font-size: 12px; width: 290px; height: 425px">Foram
lançados mais 10 modelos inéditos para o 3D BLAH.
Confira!

```

O 3D BLAH está resevando várias novidades nesta pascoa para você.
Aguarde!</textarea>

```

</td>
</tr>
</table></td>
</tr>
</table>

```

Conheca.php

```

<table width="760" border="0" cellspacing="10" cellpadding="10">
<tr>
<td valign="top"><p align="justify"><font size="6" face="Verdana, Arial,
Helvetica, sans-serif"><strong><font size="6" face="Verdana, Arial, Helvetica,
sans-serif"><strong><font color="#FF9900"><font color="#000000">Conhecendo
o<font color="#FF9900"> </font></font></font></strong></font> <font
color="#FF9900">3D
BLAH</font></strong></font></p>

```

<p align="justify">O
 3D BLAH veio para
 revolucionar

o sistema de mensagens instantâneas, proporcionando entretenimento
 através de um ambiente 3d que simula vários lugares do
 mundo.

Você vai conhecer muita gente nova de lugares que já mais
 imaginou.</p>

<p align="justify"><font size="4" face="Verdana, Arial, Helvetica, sans-
 serif">Como

funciona isso? </p>

<p align="justify"><font size="2" face="Verdana, Arial, Helvetica, sans-
 serif">Para

ser um 3D BLAH é
 muito fácil, basta você se cadastrar aqui no site e depois
 baixar o programa em Download.</p>

<p align="justify">
 Instale-o em seu computador, depois execute o programa para fazer o seu
 login (tela login). Ao abrir a tela principal do programa, você
 encontrará a sua lista de contatos e os diversos mapas que são
 os ambientes em 3d. Selecione um deles e clique em conectar. Você
 entrará no mapa e a partir daí é só se divertir
 interagindo com as pessoas que estiverem por lá. Quando se

aproximar

de alguém o nome dele aparecerá na tela e poderão
 iniciar um conversa através de uma tela de

diálogo.</p>

<p align="justify">
 Existem duas formas de se adicionar um contato: uma é clicando
 em (ícone) para fazer uma busca e a outra é iniciar uma

conversa com algu m que n o est  na sua lista de contato e, na tela de di logo, clicar em (ícone).</p>
 <p align="justify">
 Agora fique à vontade e divirta-se! </p>
 <p align="justify">Requisitos
 m nimos </p>
 <p align="justify">Os
 requisitos m nimos s o:</p>
 <p align="justify">
 • Pentium
 III 700Mhz

 • 256Mb de mem ria RAM

 • Placa de v deo 3d

 • Conex o Banda
 Larga.

 </p></td>
 </tr>
 </table>

Cadastro.php

```
<script type="text/javascript">
var erro = new String;
//Fun o de filtragem de caracteres especiais usado pelo
protocolo<strong></strong>
function Caracter(keypress){
```



```

        if (event.keyCode == 177 || event.keyCode == 172 || event.keyCode ==
162){
            alert("Caracter inválido!");
            event.returnValue = false;
        }
    }
//Função para impedir que entre caracteres diferentes de números
function Numero(keypress){
    if (event.keyCode > 57 || event.keyCode < 48){
        event.returnValue = false;
    }
}
//Função para mascar os campos que contenham telefone
function MascaraTEL(keypress, objeto){
    if (event.keyCode > 57 || event.keyCode < 48){
        event.returnValue = false;
    }
    campo = eval (objeto);
    if (campo.value.length == 0){
        campo.value = campo.value + "(";
    }else if (campo.value.length == 3){
        campo.value = campo.value + ")";
    }else if (campo.value.length == 8){
        campo.value = campo.value + "-";
    }
}
//Função para validar o campo E-mail
function VerEmail(){
    var email = document.cadastro.email.value;
    var verarroba = email.indexOf("@");
    if(verarroba == -1){

```

```

        erro += "Informar o sinal de arroba '@' no USUÁRIO \n";
    }else{
        var splits = email.split("@",2);
        var nome = splits[0];
        var dominio = splits[1];
        var verponto = dominio.indexOf(".");
        if(dominio != ""){
            if(verponto != -1){
                var teste = dominio.split(".",2);
                var nomedominio = teste[0];
                var extensao = teste[1];

                if (extensao == "")
                    erro += "Informe a extensão do USUÁRIO \n";
            }else{
                erro += "Informe a extensão do USUÁRIO \n";
            }
        }else{
            erro += "Informe o domínio e a extensão do USUÁRIO \n";
        }
    }
}

//Função para confirmar se todos os campos obrigatórios foram incluídos
function Confirma(){
    if (document.cadastro.nome.value == ""){
        erro += "Preencha o campo NOME para efetuar o cadastro \n";
    }

    if (document.cadastro.dia.value == "" || document.cadastro.mes.value == ""
    || document.cadastro.ano.value == ""){

```

```
        erro += "Preencha o campo DATA NASCIMENTO para efetuar o
cadastro \n";
    }

    if (document.cadastro.apelido.value == ""){
        erro += "Preencha o campo NICK para efetuar o cadastro \n";
    }

    if (document.cadastro.email.value == ""){
        erro += "Preencha o campo USUÁRIO para efetuar o cadastro \n";
    }else{
        VerEmail();
    }

    if(document.cadastro.senha.value == ""){
        erro += "Preencha o campo SENHA para efetuar o cadastro \n";
    }

    if(document.cadastro.testesenha.value == ""){
        erro += "Preencha o campo CONFIRMAR SENHA para efetuar o
cadastro \n";
    }

    if(document.cadastro.senha.value != document.cadastro.testesenha.value){
        erro += "Preencha o campo CONFIRMAR SENHA com o mesmo
valor de SENHA \n";
    }

    if(document.cadastro.cidade.value == ""){
        erro += "Preencha o campo CIDADE para efetuar o cadastro \n";
```

```

}

if(document.cadastro.uf.value == ""){
    erro += "Preencha o campo ESTADO para efetuar o cadastro \n";
}

if(erro.length > 0){
    alert(erro);
    erro = "";
    return false;
}else{
    return true;
}
}
}

</script>

<table width="760" border="0" cellspacing="20" cellpadding="0">
<tr>
<td valign="top">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
<form name="cadastro" action="tratacad.php" method="post"
onSubmit="return Confirma()">
<tr>
<td colspan="2"><div align="left"><font size="4" face="Verdana, Arial,
Helvetica, sans-serif"><strong><font size="6">Cadastro
<font color="#FF9900">3D BLAH</font></font><font color="#FF9900">
</font></strong></font></div></td>
<td width="19%" height="5" align="right"> </td>
<td width="81%"> </td>

```

```

</tr>

    <?php
        //Mensagem de erro caso já exista um registro com o usuário
informado

        $ver = $_GET['ver'];
        if($ver != ""){
            print("<tr bgcolor=\"#CC0000\">
                <td height=\"25\" colspan=\"2\"><strong><font color=\"#FFFFFF\" size=\"2\"
face=\"Verdana, Arial, Helvetica, sans-serif\">&nbsp;N&atilde;o
foi possivel se cadastrar. J&aacute; h&aacute; um registro com esse
Usu&aacute;rio</font></strong></td>
            </tr>");
        }
    ?>

<tr>
    <td align="right"><div align="left"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif"><strong><font color="#FF9900">Dados
    pessoais</font></strong> &nbsp;</font></div></td>
    <td align="right"> <font color="#FF9900" size="2" face="Verdana, Arial,
Helvetica, sans-serif">*campo
    obrigat&oacute;rio</font></td>
</tr>

<tr>
    <td align="right" bgcolor="eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Nome
        completo</font><font color="#FF9900" size="2" face="Verdana, Arial,
Helvetica, sans-serif">*</font><font size="2" face="Verdana, Arial, Helvetica, sans-
serif">:</font></td>
        <td><input name="nome" type="text" id="nome" size="50"
onKeyPress="Caracter(window.event.keyCode);"></td>
</tr>

```

```

<tr>
  <td align="right" bgcolor="eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Data
    nascimento</font><font color="#FF9900" size="2" face="Verdana, Arial,
Helvetica, sans-serif">*</font><font size="2" face="Verdana, Arial, Helvetica, sans-
serif">:</font></td>
    <td><input name="dia" type="text" id="dia2"
onKeyPress="Numero(window.event.keyCode);" size="3" maxlength="2">
    /
    <input name="mes" type="text" id="mes"
onKeyPress="Numero(window.event.keyCode);" size="3" maxlength="2">
    /
    <input name="ano" type="text" id="ano"
onKeyPress="Numero(window.event.keyCode);" size="6" maxlength="4">
    <font color="#999999" size="1" face="Verdana, Arial, Helvetica, sans-
serif">dd/mm/aaaa</font>
  </td>
</tr>
<tr>
  <td align="right" bgcolor="eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Sexo:</font></td>
    <td><input type="radio" name="sexo" value="1"> <font size="2"
face="Verdana, Arial, Helvetica, sans-serif">masculino
    <input type="radio" name="sexo" value="0">
    feminino</font></td>
</tr>
<tr>
  <td align="right" bgcolor="eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Estado
    civil:</font></td>
    <td><select name="estCivil" id="estCivil">

```

```

        <option value="-1"></option>
        <option value="0">Solteiro</option>
        <option value="1">Namorando</option>
        <option value="2">Casado</option>
        <option value="3">Separado</option>
        <option value="4">Divorciado</option>
        <option value="5">Vi&uacute;vo</option>
        <option value="6">Outro</option>
    </select></td>
</tr>
<tr>
    <td height="5" align="right"> </td>
    <td> </td>
</tr>
<tr align="left">
    <td colspan="2"><font size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong><font color="#FF9900">Dados
    3D BLAH</font></strong>&nbsp;</font></td>
</tr>
<tr>
    <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Nick
        <font size="1">(apelido)</font><font color="#FF9900" size="2"
face="Verdana, Arial, Helvetica, sans-serif">*</font>:</font></td>
    <td><input name="apelido" type="text" id="apelido" size="30"
onKeyPress="Caracter(window.event.keyCode);">
        <font color="#999999" size="1" face="Verdana, Arial, Helvetica, sans-
serif">letras
            min&uacute;culas, ex: joaopaulo</font> </td>
</tr>
<tr>

```

```

        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
        Helvetica, sans-serif">Usu&aacute;rio
        <font size="1">(e-mail)</font><font color="#FF9900" size="2"
        face="Verdana, Arial, Helvetica, sans-serif">*</font>:
        </font></td>

        <td><input name="email" type="text" id="email" size="30"
        onKeyPress="Caracter(window.event.keyCod);">
        <font color="#999999" size="1" face="Verdana, Arial, Helvetica, sans-
        serif">ex:
        nome@dominio.com.br </font> </td>
    </tr>
    <tr>
        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
        Helvetica, sans-serif">Senha</font><font color="#FF9900" size="2"
        face="Verdana, Arial, Helvetica, sans-serif">*</font><font size="2" face="Verdana,
        Arial, Helvetica, sans-serif">:</font></td>
        <td><input name="senha" type="password" id="senha" size="15"
        maxlength="8" onKeyPress="Caracter(window.event.keyCod);">
        <font color="#999999" size="1" face="Verdana, Arial, Helvetica, sans-
        serif">m&aacute;ximo
        de 8 caracteres</font> </td>
    </tr>
    <tr>
        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
        Helvetica, sans-serif">Confrimar
        Senha</font><font color="#FF9900" size="2" face="Verdana, Arial,
        Helvetica, sans-serif">*</font><font size="2" face="Verdana, Arial, Helvetica, sans-
        serif">:</font></td>
        <td><input name="testesenha" type="password" id="testesenha2"
        size="15" onKeyPress="Caracter(window.event.keyCod);">

```



```

        <font color="#999999" size="1" face="Verdana, Arial, Helvetica, sans-
        serif">inserir
            a mesma senha acima para fazer a confrima&ccedil;&atilde;o</font>
        </td>
    </tr>
    <tr>
        <td align="right" height="5"> </td>
        <td> </td>
    </tr>
    <tr>
        <td colspan="2" align="right"><div align="left"><font size="2"
        face="Verdana, Arial, Helvetica, sans-serif"><strong><font
        color="#FF9900">Dados
            adicionais </font></strong>&nbsp;</font></div></td>
    </tr>
    <tr>
        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
        Helvetica, sans-serif">Cidade</font><font color="#FF9900" size="2"
        face="Verdana, Arial, Helvetica, sans-serif">*</font><font size="2" face="Verdana,
        Arial, Helvetica, sans-serif">:</font></td>
        <td><input name="cidade" type="text" id="cidade" size="30"
        onKeyPress="Caracter(window.event.keyCod);"></td>
    </tr>
    <tr>
        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
        Helvetica, sans-serif">Estado</font><font color="#FF9900" size="2"
        face="Verdana, Arial, Helvetica, sans-serif">*</font><font size="2" face="Verdana,
        Arial, Helvetica, sans-serif">:</font></td>
        <td><input name="uf" type="text" id="uf" size="30"
        onKeyPress="Caracter(window.event.keyCod);">
        </td>
    </tr>

```

```
</tr>
```

```
<tr>
```

```
    <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Pa&iacute;s</font><font color="#FF9900" size="2"
face="Verdana, Arial, Helvetica, sans-serif">*</font><font size="2" face="Verdana,
Arial, Helvetica, sans-serif">:</font></td>
```

```
    <td> <select name="pais" id="pais">
```

```
      <option value="-1"></option>
```

```
      <option value="0">Afeganist&atilde;o</option>
```

```
      <option value="2">&Aacute;frica do Sul</option>
```

```
      <option value="3">Alb&acirc;nia</option>
```

```
      <option value="4">Alemanha</option>
```

```
      <option value="5">Andorra</option>
```

```
      <option value="6">Angola</option>
```

```
      <option value="7">Ant&iacute;gua e Barbuda</option>
```

```
      <option value="8">Ar&aacute;bia Saudita</option>
```

```
      <option value="9">Arg&eacute;lia</option>
```

```
      <option value="10">Argentina</option>
```

```
      <option value="11">Arm&eacute;nia</option>
```

```
      <option value="12">Austr&aacute;lia</option>
```

```
      <option value="13">&Aacute;ustria</option>
```

```
      <option value="14">Azerbaij&atilde;o</option>
```

```
      <option value="15">Baamas</option>
```

```
      <option value="16">Bangladeche</option>
```

```
      <option value="17">Barbados</option>
```

```
      <option value="18">Bar&eacute;m</option>
```

```
      <option value="19">B&eacute;lgica</option>
```

```
      <option value="20">Belize</option>
```

```
      <option value="21">Benim</option>
```

```
      <option value="22">Bielorr&uacute;ssia</option>
```

```
      <option value="23">Birm&acirc;nia</option>
```

<option value="24">Bolívia</option>
 <option value="25">Bosnia e Herzegovina</option>
 <option value="26">Botsuana</option>
 <option value="27" selected>Brasil</option>
 <option value="28">Brunei</option>
 <option value="29">Bulgária</option>
 <option value="30">Burquina Faso</option>
 <option value="31">Burúndi</option>
 <option value="32">Butão</option>
 <option value="33">Cabo Verde</option>
 <option value="34">Camarões</option>
 <option value="35">Camboja</option>
 <option value="36">Canadá</option>
 <option value="37">Cazaquistão</option>
 <option value="38">Centro-Africana (Rep.)</option>
 <option value="39">Chade</option>
 <option value="40">Checa (Rep.)</option>
 <option value="41">Chile</option>
 <option value="42">China</option>
 <option value="43">Chipre</option>
 <option value="44">Colômbia</option>
 <option value="45">Comores</option>
 <option value="46">Congo</option>
 <option value="47">Congo (Rep. Dom.)</option>
 <option value="48">Coreia do Norte</option>
 <option value="49">Coreia do Sul</option>
 <option value="50">Costa do Marfim</option>
 <option value="51">Costa Rica</option>
 <option value="52">Croácia</option>
 <option value="53">Cuba</option>
 <option value="54">Dinamarca</option>

<option value="55">Djibouti</option>
<option value="56">Domínica</option>
<option value="57">Dominicana (Rep.)</option>
<option value="58">Egipto</option>
<option value="59">El Salvador</option>
<option value="60">Emiratos Árabes Unidos</option>
<option value="61">Equador</option>
<option value="62">Eritreia</option>
<option value="63">Eslováquia</option>
<option value="64">Eslovénia</option>
<option value="65">Espanha</option>
<option value="66">Est. Fed. Micronésia</option>
<option value="67">Estados Unidos</option>
<option value="68">Estónia</option>
<option value="69">Etiópia</option>
<option value="70">Fiji</option>
<option value="71">Filipinas</option>
<option value="72">Finlândia</option>
<option value="73">França</option>
<option value="74">Gabão</option>
<option value="75">Gâmbia</option>
<option value="76">Gana</option>
<option value="77">Geórgia</option>
<option value="78">Granada</option>
<option value="79">Grécia</option>
<option value="80">Guatemala</option>
<option value="81">Guiana</option>
<option value="82">Guiné,</option>
<option value="83">Guiné Equatorial</option>
<option value="84">Guiné-Bissau</option>
<option value="85">Haiti</option>

<option value="86">Holanda</option>
<option value="87">Honduras</option>
<option value="88">Hungria</option>
<option value="89">Iémen</option>
<option value="90">Ilhas Marshall</option>
<option value="91">Ilhas Salomão</option>
<option value="92">Índia</option>
<option value="93">Indonésia</option>
<option value="94">Irãe</option>
<option value="95">Iraque</option>
<option value="96">Irlanda</option>
<option value="97">Islândia</option>
<option value="98">Israel</option>
<option value="99">Itália</option>
<option value="100">Jamaica</option>
<option value="101">Japão</option>
<option value="102">Jordânia</option>
<option value="103">Kiribati</option>
<option value="104">Kuwait</option>
<option value="105">Laos</option>
<option value="106">Lesoto</option>
<option value="107">Letónia</option>
<option value="108">Líbano</option>
<option value="109">Libéria</option>
<option value="110">Líbia</option>
<option value="111">Liechtenstein</option>
<option value="112">Lituânia</option>
<option value="113">Luxemburgo</option>
<option value="114">Macedónia</option>
<option value="115">Madagáscar</option>
<option value="116">Malásia</option>

<option value="117">Malávi</option>
 <option value="118">Maldivas</option>
 <option value="119">Mali</option>
 <option value="120">Malta</option>
 <option value="121">Marrocos</option>
 <option value="122">Maurícia</option>
 <option value="123">Mauritânia</option>
 <option value="124">México</option>
 <option value="125">Moçambique</option>
 <option value="126">Moldávia</option>
 <option value="127">Mónaco</option>
 <option value="128">Mongólia</option>
 <option value="129">Namíbia</option>
 <option value="130">Nauru</option>
 <option value="131">Nepal</option>
 <option value="132">Nicarágua</option>
 <option value="133">Níger</option>
 <option value="134">Nigéria</option>
 <option value="135">Noruega</option>
 <option value="136">Nova Zelândia</option>
 <option value="137">Omã;</option>
 <option value="138">Palau</option>
 <option value="139">Panamá;</option>
 <option value="140">Papua-Nova Guiné;</option>
 <option value="141">Paquistão</option>
 <option value="142">Paraguai</option>
 <option value="143">Peru</option>
 <option value="144">Polónia</option>
 <option value="145">Portugal</option>
 <option value="146">Qatar</option>
 <option value="147">Quênia</option>

<option value="148">Quirguistão</option>
 <option value="149">Reino Unido</option>
 <option value="150">Roménia</option>
 <option value="151">Ruanda</option>
 <option value="152">Rússia</option>
 <option value="153">Saara Ocidental</option>
 <option value="154">Saint Kittis e Nevis</option>
 <option value="155">Samoa</option>
 <option value="156">San Marinho</option>
 <option value="157">Santa Lúcia</option>
 <option value="158">São Tomé e
 Príncipe</option>
 <option value="159">São Vicente e Granadinas</option>
 <option value="160">Senegal</option>
 <option value="161">Servia e Montenegro</option>
 <option value="162">Serra Leoa</option>
 <option value="163">Seychelles</option>
 <option value="164">Singapura</option>
 <option value="165">Somália</option>
 <option value="166">Sri Lanca</option>
 <option value="167">Suazilândia</option>
 <option value="168">Sudão</option>
 <option value="169">Suécia</option>
 <option value="170">Suí ça</option>
 <option value="171">Suriname</option>
 <option value="172">Syria</option>
 <option value="173">Tadjiquistão</option>
 <option value="174">Tailândia</option>
 <option value="175">Taiwan</option>
 <option value="176">Tanzânia</option>
 <option value="177">Timor Leste</option>

```

        <option value="178">Togo</option>
        <option value="179">Tonga</option>
        <option value="180">Trindade e Tobago</option>
        <option value="181">Tun&iacute;sia</option>
        <option value="182">Turquemenist&atilde;o</option>
        <option value="183">Turquia</option>
        <option value="184">Tuvalu</option>
        <option value="185">Ucr&acirc;nia</option>
        <option value="186">Uganda</option>
        <option value="187">Uruguai</option>
        <option value="188">Usbequist&atilde;o</option>
        <option value="189">Vanuatu</option>
        <option value="190">Vaticano</option>
        <option value="191">Venezuela</option>
        <option value="192">Vietnam</option>
        <option value="193">Z&acirc;mbia</option>
        <option value="194">Zimbabu&eacute;</option>
    </select></td>
</tr>
<tr>
    <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Telefone:</font></td>
    <td> <font size="1" face="Verdana, Arial, Helvetica, sans-serif">
        <input name="telefone" type="text" id="telefone"
onKeyPress="MascaraTEL(window.event.keyCode, this);" size="15"
maxlength="13">
        </font><font color="#999999" size="1" face="Verdana, Arial, Helvetica,
sans-serif">ex:
        (22)2222-2222</font></td>
</tr>
<tr>

```



```

        <td align="right" bgcolor="#eeeeee"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Celular:</font></td>
        <td><font size="1" face="Verdana, Arial, Helvetica, sans-serif">
        <input name="celular" type="text" id="celular" size="15"
onKeyPress="MascaraTEL(window.event.keyCode, this);" maxlength="13">
        </font><font color="#999999" size="1" face="Verdana, Arial, Helvetica,
sans-serif">ex:
        (22)9999-9999</font></td>
    </tr>
    <tr>
        <td align="right"><font size="2" face="Verdana, Arial, Helvetica, sans-
serif">&nbsp;</font></td>
        <td><input type="submit" name="Submit" value="Cadastrar"> <input
type="reset" name="Submit2" value="Apagar">
        <input type="button" name="Submit22" value="Cancelar"> </td>
    </tr>
</form>
</table></td>
</tr>
</table>

```

Tratacad.php

```

<?php
//Chama a página conect.php
require_once("conect.php");

//Verifica se já existe o usuário informado e caso exista redireciona para a página
cadastro para mostrar uma mensagem de erro
$testaemail = $_POST['email'];

```

```

$verEmail = mssql_query("SELECT ChaveUsuario FROM Usuario WHERE Email
= '$testaemail'");
if(mssql_num_rows($verEmail) > 0)
    header('Location:index.php?pagina=cadastro&ver=1');

//Verifica o último registro
$pesquisa = mssql_query("SELECT MAX(ChaveUsuario) AS id FROM Usuario");
$dados = mssql_fetch_array($pesquisa);
$quant = $dados['id'];

//RECEBE OS DADOS DO FORMULÁRIO
$id = $quant + 1;
$nome = $_POST['nome'];
$apelido = $_POST['apelido'];
$nascimento .= $_POST['mes'];
$nascimento .= "/";
$nascimento .= $_POST['dia'];
$nascimento .= "/";
$nascimento .= $_POST['ano'];
$sexo = $_POST['sexo'];
$cidade = $_POST['cidade'];
$uf = $_POST['uf'];
$pais = $_POST['pais'];
$email = $_POST['email'];
$senha = $_POST['senha'];
$estCivil = $_POST['estCivil'];
$telefone = $_POST['telefone'];
$celular = $_POST['celular'];

```

```
//insere dados
$dados = "INSERT INTO Usuario (ChaveUsuario, Nome, Apelido, Nascimento,
Sexo, Cidade, Estado, Pais, Email, Senha, ChaveModelo, ChaveServidor,
StatusUsuario, Foto, EstadoCivil, Telefone, Celular) values
('$id','$nome','$apelido','$nascimento','$sexo','$cidade','$uf','$pais','$email','$senha'
, ' ', ' ', ' ', ' ', '$estCivil', '$telefone', '$celular')";
$sql = mssql_query($dados);
if($sql){
    header('Location:index.php?pagina=cadok');
}else{
    header('Location:index.php?pagina=cadnot');
}

?>
```

Connect.php

```
<?php
//Dados do servidor
$hostname = 'localhost';
$user = 'sa';
$password = '';
$databse = 'blah';

//Conexao com o banco de dados MSSQL
$conexao = mssql_connect($hostname, $user, $password) or die ("Não foi
possivel estabelecer conexão com o Banco de Dados!");
mssql_select_db($databse) or die ("Não foi possivel encontrar a tebela!");

?>
```

Cadnot.php

```

<table width="760" border="0" cellspacing="20" cellpadding="0">
  <tr>
    <td height="270" align="center" valign="middle"> <table width="500" border="0"
cellspacing="0" cellpadding="0">
      <tr>
        <td width="127"></td>
        <td width="10" bgcolor="#FF9900"> </td>
        <td align="center" bgcolor="#eeeeee"><p><font color="#000000" size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong>Cadastro
n&atilde;o foi possivel ser efetuado!</strong></font></p>
<p><strong><font color="#000000" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><a href="index.php" style="color: #000000">P&aacute;gina
inicial</a></font></strong></p></td>
        <td width="10" bgcolor="#FF9900"> </td>
      </tr>
    </table></td>
  </tr>
</table>

```

Cadok.php

```

<table width="760" border="0" cellspacing="20" cellpadding="0">
  <tr>
    <td height="270" align="center" valign="middle"> <table width="500" border="0"
cellspacing="0" cellpadding="0">
      <tr>
        <td width="127"></td>
        <td width="10" bgcolor="#FF9900"> </td>

```

```

        <td align="center" bgcolor="#eeeeee"><p><font color="#000000" size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong>Cadastro
        efetuado com sucesso!</strong></font></p>
        <p><strong><font color="#000000" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><a href="index.php?pagina=adquirira" style="color:
#000000">Baixar
        o programa</a></font></strong></p></td>
        <td width="10" bgcolor="#FF9900"> </td>
    </tr>
</table></td>
</tr>
</table>

```

Adquirira.php

```

<script type="text/javascript">
//Função que confere se os dados foram incluídos
function Conf(){
    if(document.login.usuario.value != ""){
        if(document.login.senha.value != ""){
            return true;
        }
        else{
            alert("Preencha o campo SENHA!")
            return false;
        }
    }
    else{
        alert("Preencha o campo USUÁRIO!")
        return false;
    }
}

```

```

</script>
</head>

<body>
<table width="760" border="0" cellspacing="10" cellpadding="0">
  <tr>
    <td><table width="100%" border="0" cellspacing="10" cellpadding="0">
      <tr>
        <td width="355" valign="top">
          <p><font size="6" face="Verdana, Arial, Helvetica, sans-serif"><strong>J&acute;
            sou <font color="#FF9900">3D BLAH</font></strong></font></p>
          <p><font color="#000000" size="2" face="Verdana, Arial, Helvetica, sans-
            serif">Agora
            para adquirir o seu 3D BLAH basta voc&ecirc; entrar com o seu <font
            color="#FF9900">usu&acute;rio</font>
            e <font color="#FF9900">senha <font color="#333333">que registrou
            em seu cadastro.</font></font></font></p>
          <table width="100%" border="0" cellpadding="0" cellspacing="5"
            bgcolor="#eeeeee">
            <form action="autentica.php" name="login" method="post"
            onSubmit="return Conf()">
              <tr>
                <td width="21%" align="right"><font color="#000000" size="2"
                face="Verdana, Arial, Helvetica, sans-
                serif"><strong>usu&acute;rio:</strong></font></td>
                <td width="79%"><input name="email" type="text" id="usuario3"
                size="27"></td>
              </tr>
              <tr>
                <td align="right"><font color="#000000" size="2" face="Verdana, Arial,
                Helvetica, sans-serif"><strong>senha:</strong></font></td>

```

```

        <td><input name="senha" type="password" id="senha" size="27"></td>
    </tr>
    <tr>
        <td align="right">&nbsp;</td>
        <td><input type="submit" name="Submit" value="ok"></td>
    </tr>
</form>
</table>

</td>
<td width="1" bgcolor="#CCCCCC"> </td>
<td valign="top"><p><font size="6" face="Verdana, Arial, Helvetica, sans-
serif"><strong>Quero
    ser <font color="#FF9900">3D BLAH</font></strong></font></p>
    <p><font color="#000000" size="2" face="Verdana, Arial, Helvetica, sans-
serif">Para
        voc&ecirc; se tornar uma 3D BLAH basta voc&ecirc; fazer o seu cadastro.
        Depois baixe o programa para instalar em computador.</font></p>
    <p><font color="#FF9900" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong><a href="index.php?pagina=cadastro" style="color:
#FF9900">Quero
        me cadastrar agora mesmo!</a></strong></font></p></td>
</tr>
</table></td>
</tr>
</table>

```

Autentica.php

```

<?php
//Chama a página conect.php

```

```

require_once("conect.php");

//Recebe os dados do formulário
$email = $_POST['email'];
$senha = $_POST['senha'];

//Pesquisa se o usuário e a senha estão corretos
$pesquisa = mssql_query("SELECT * FROM Usuario WHERE Email = '$email'
AND Senha = '$senha'");
$row = mssql_num_rows($pesquisa);
if($row == 0){
    header("Location: index.php?pagina=adiquiraerro");
    echo "Não foi possível encontrar";
}
else {
    //Pega os dados do usuário
    while($dados = mssql_fetch_array($pesquisa)){
        $id = $dados['ChaveUsuario'];
        $nome = $dados['Apelido'];
    }

    //inicia a seção para segurança
    session_start();

    $_SESSION['idblah'] = $id;
    $_SESSION['nomeblah'] = $nome;

    //REDIRECIONA PARA A PÁGINA QUE VAI EXIBIR OS PRODUTOS
    header("Location: download.php");
}

```


?>

Adiquiraerro.php

```
<table width="760" border="0" cellspacing="20" cellpadding="0">
  <tr>
    <td height="270" align="center" valign="middle"> <table width="500" border="0"
cellspacing="0" cellpadding="0">
      <tr>
        <td width="127"></td>
        <td width="10" bgcolor="#FF9900"> </td>
        <td align="center" bgcolor="#eeeeee"><p><font color="#000000" size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong>N&atilde;o
foi poss&iacute;vel encontrar
usu&aacute;rio.</strong></font><strong><font color="#000000" size="2"
face="Verdana, Arial, Helvetica, sans-serif">
      <br>
      Usu&aacute;rio e/ou Senha incorreto.</font></strong></p>
      <p><strong><font color="#000000" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><a href="index.php?pagina=adquira" style="color:
#000000">P&aacute;gina
de login</a></font></strong></p></td>
        <td width="10" bgcolor="#FF9900"> </td>
      </tr>
    </table></td>
  </tr>
</table>
```

Download.php

```

<?php
//Verifica se o usuário está logado se não estiver redireciona para o login
session_start();

if (!isset($_SESSION['idblah'])){
header("Location: index.php?pagina=adquira");
}
else{
print("<html>
<head>
<title>| 3D BLAH |</title>
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">
</head>

<body bottommargin=\"0\" leftmargin=\"0\" topmargin=\"0\" rightmargin=\"0\">
<table width=\"100%\" height=\"100%\" border=\"0\" cellpadding=\"0\"
cellspacing=\"0\">
<tr>
<td align=\"center\" valign=\"top\" background=\"imagens/fundo.gif\">
<table width=\"760\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\">
<tr>
<td height=\"25\">&nbsp;</td>
</tr>
</table>
<table width=\"760\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\">
<tr>
<td><object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\" width=\"760\" height=\"128\">
<param name=\"movie\" value=\"topo.swf\">
<param name=\"quality\" value=\"high\">

```

```

        <embed src=\"topo.swf\" quality=\"high\"
pluginspage=\"http://www.macromedia.com/go/getflashplayer\"
type=\"application/x-shockwave-flash\" width=\"760\"
height=\"128\"></embed></object></td>
    </tr>
</table>");
    if(isset($_SESSION['idblah'])){
        $nome = $_SESSION['nomeblah'];
        print("<table width=\"760\" border=\"0\" cellpadding=\"7\" background=\"imagens/corpo.gif\">
    <tr>
        <td height=\"15\" valign=\"top\">
            <table width=\"100%\" border=\"0\" cellpadding=\"2\">
                <tr>
                    <td bgcolor=\"#eeeeee\"
align=\"right\"><font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>$nome</b>, você está logado. <a href=\"logout.php\" style=\"color:
#000000\"><b>Sair</b></a>&nbsp;&nbsp;&nbsp;</font></td>
                </tr>
            </table>
        </td>
    </tr>
</table>");
    }
    print("<table width=\"760\" border=\"0\" cellpadding=\"0\">
    <tr>
        <td height=\"270\" align=\"left\" valign=\"top\"
background=\"imagens/corpo.gif\">

            <table width=\"100%\" border=\"0\" cellpadding=\"15\"
cellpadding=\"0\">

```

```

<tr>
  <td width="50%">
<table width="100%" border="0" cellspacing="1" cellpadding="5">
  <tr>
    <td colspan="2" bgcolor="#FF9900"><strong><font color="#FFFFFF"
size="4" face="Verdana, Arial, Helvetica, sans-serif">3D
      BLAH </font></strong></td>
  </tr>
  <tr>
    <td width="50%" rowspan="2"><a href="3dblah.zip"></a></td>
    <td valign="top" bgcolor="eeeeee"><font size="2" face="Verdana,
Arial, Helvetica, sans-serif">Baixe
      o 3D BLAH em seu pc e divirta-se conhecendo pessoas
      novas.</font></td>
  </tr>
  <tr>
    <td height="5" valign="bottom" bgcolor="eeeeee"><font size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong><font
color="ff9900">Total:</font>
      1 </strong></font></td>
  </tr>
</table></td>
<td><table width="100%" border="0" cellspacing="1" cellpadding="5">
  <tr>
    <td colspan="2" bgcolor="#FF9900"><strong><font color="#FFFFFF"
size="4" face="Verdana, Arial, Helvetica, sans-
serif">Atualiza&ccedil;&otilde;es</font></strong></td>
  </tr>
  <tr>

```

```

        <td width="50%" rowspan="2"><a href="atualizacao.php"></td>
        <td valign="top" bgcolor="eeeeee"><font size="2" face="Verdana,
Arial, Helvetica, sans-serif">Instale
        as atualizações para melhorar o seu 3D BLAH.<br>
        </font></td>
</tr>
<tr>
        <td height="5" valign="bottom" bgcolor="eeeeee"><font size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong><font
color="ff9900">Total:</font>
        0</strong></font></td>
</tr>
</table></td>
</tr>
<tr>
        <td><table width="100%" border="0" cellspacing="1" cellpadding="5">
        <tr>
                <td colspan="2" bgcolor="#FF9900"><strong><font color="#FFFFFF"
size="4" face="Verdana, Arial, Helvetica, sans-
serif">Mapas</font></strong></td>
        </tr>
        <tr>
                <td width="50%" rowspan="2"><a href="mapa.php"></a></td>
                <td valign="top" bgcolor="eeeeee"><font size="2" face="Verdana,
Arial, Helvetica, sans-serif">Incremente
                o seu 3D BLAH com os mapas mais alucinantes e você irá
                conhecer novos lugares.</font></td>
        </tr>
        <tr>

```

```

        <td height=\"5\" valign=\"bottom\" bgcolor=\"eeeeee\"><font size=\"2\"
face=\"Verdana, Arial, Helvetica, sans-serif\"><strong><font
color=\"ff9900\">Total:</font></strong></td>

        require_once("conect.php");
        $servidor = mssql_query("SELECT * FROM
Servidor");

        $quantservidor = mssql_num_rows($servidor);
        print(" $quantservidor");
        print(</strong></font></td>

    </tr>
</table></td>

<td><table width=\"100%\" border=\"0\" cellspacing=\"1\" cellpadding=\"5\">
    <tr>
        <td colspan=\"2\" bgcolor=\"#FF9900\"><strong><font color=\"#FFFFFF\"
size=\"4\" face=\"Verdana, Arial, Helvetica, sans-
serif\">Modelos</font></strong></td>
    </tr>
    <tr>
        <td width=\"50%\" rowspan=\"2\"><a href=\"modelo.php\"><img
src=\"imagens/modelo.jpg\" width=\"96\" height=\"100\" border=\"0\"></a></td>
        <td valign=\"top\" bgcolor=\"eeeeee\"><font size=\"2\" face=\"Verdana,
Arial, Helvetica, sans-serif\">Mude
        de cara no 3D BLAH e seja quem voc&ecirc; sempre quis
ser.</font></td>
    </tr>
    <tr>
        <td height=\"5\" valign=\"bottom\" bgcolor=\"eeeeee\"><font size=\"2\"
face=\"Verdana, Arial, Helvetica, sans-serif\"><strong><font
color=\"ff9900\">Total:</font></strong></td>
        require_once("conect.php");

```

```

$modelo = mssql_query("SELECT * FROM
Modelo");

$quantmodelo = mssql_num_rows($modelo);
print(" $quantmodelo");
print("</strong></font></td>

</tr>
</table></td>
</tr>
</table>

</td>
</tr>
</table>
<table width=\"760\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\">
<tr>
<td><object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\" width=\"760\" height=\"100\">
<param name=\"movie\" value=\"rodape.swf\">
<param name=\"quality\" value=\"high\">
<embed src=\"rodape.swf\" quality=\"high\"
pluginspage=\"http://www.macromedia.com/go/getflashplayer\"
type=\"application/x-shockwave-flash\" width=\"760\"
height=\"100\"></embed></object></td>
</tr>
</table></td>
</tr>
</table>
</body>
</html>");
}

```

?>

Atualizacao.php

```
<?php
//Verifica se o usuário está logado se não estiver redireciona para o login
session_start();

if (!isset($_SESSION['idblah'])){
header("Location: index.php?pagina=adquira");
}
else{
print("<html>
<head>
<title>| 3D BLAH |</title>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>
</head>

<body bottommargin='0' leftmargin='0' topmargin='0' rightmargin='0'>
<table width='100%' height='100%' border='0' cellpadding='0'
cellspacing='0'>
<tr>
<td align='center' valign='top' background='imagens/fundo.gif'>
<table width='760' border='0' cellspacing='0' cellpadding='0'>
<tr>
<td height='25'>&nbsp;</td>
</tr>
</table>
<table width='760' border='0' cellspacing='0' cellpadding='0'>
<tr>
```



```

<td height=\ "270\ " align=\ "left\ " valign=\ "top\ "
background=\ "imagens/corpo.gif\ ">
    <table width=\ "100%\ " border=\ "0\ " cellpadding=\ "15\ ">
        <tr background=\ "imagens/corpo.gif\ ">
            <td>
                <font size=\ "2\ " face=\ "Verdana, Arial, Helvetica, sans-
                serif\ "><strong>Instru&ccedil;&otilde;es:</strong> Ap&ocirc;e; baixar o
                arquivo em seu pc extraia-o na pasta onde o 3D BLAH est&aacute;e;
                instalado.</font>
            </td>
        </tr>
        <tr background=\ "imagens/corpo.gif\ "><td align=\ "left\ "><table
        width=\ "100%\ " border=\ "0\ " cellpadding=\ "5\ ">
            <tr><td bgcolor=\ "#ff9900\ ">
                <font color=\ "#ffffff\ " size=\ "2\ " face=\ "Verdana, Arial, Helvetica,
                sans-serif\ "><b>Atualiza&ccedil;es</b></font></tr></td>
                <tr><td bgcolor=\ "#eeeeee\ "><font color=\ "#cc0000\ " size=\ "2\ "
                face=\ "Verdana, Arial, Helvetica, sans-serif\ "><b>N&acirc;o h&acirc;a
                atualiza&ccedil;es</b></font></tr></td>
                <tr><td bgcolor=\ "#ffffff\ " align=\ "right\ "><font size=\ "2\ "
                face=\ "Verdana, Arial, Helvetica, sans-serif\ "><a href=\ "download.php\ "
                style=\ "color: 000000\ "><b>Voltar</b><a></font></tr></td>
            </table></td> </tr>
        </table>
    </td>
</tr>
</table>
<table width=\ "760\ " border=\ "0\ " cellpadding=\ "0\ " cellspacing=\ "0\ ">
    <tr>

```

```

        <td><object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\" width=\"760\" height=\"100\">
        <param name=\"movie\" value=\"rodape.swf\">
        <param name=\"quality\" value=\"high\">
        <embed src=\"rodape.swf\" quality=\"high\"
pluginspage=\"http://www.macromedia.com/go/getflashplayer\"
type=\"application/x-shockwave-flash\" width=\"760\"
height=\"100\"></embed></object></td>
    </tr>
</table></td>
</tr>
</table>
</body>
</html>");
}
?>

```

Mapa.php

```

<?php
//Verifica se o usuário está logado se não estiver redireciona para o login
session_start();

if (!isset($_SESSION['idblah'])){
header("Location: index.php?pagina=adquira");
}
else{
print("<html>
<head>
<title>| 3D BLAH |</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bottommargin="0" leftmargin="0" topmargin="0" rightmargin="0">
<table width="100%" height="100%" border="0" cellpadding="0"
cellspacing="0">
  <tr>
    <td align="center" valign="top" background="imagens/fundo.gif">
      <table width="760" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td height="25">&nbsp;</td>
        </tr>
      </table>
      <table width="760" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0" width="760" height="128">
          <param name="movie" value="topo.swf">
          <param name="quality" value="high">
          <embed src="topo.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash" width="760"
height="128"></embed></object></td>
        </tr>
      </table>");
      if(isset($_SESSION['idblah'])){
        $nome = $_SESSION['nomeblah'];
        print("<table width="760" border="0" cellspacing="0"
cellpadding="7" background="imagens/corpo.gif">
        <tr>

```


arquivo em seu pc extraia-o na pasta onde o 3D BLAH está instalado.

</td>

</tr>

<tr background=\"imagens/corpo.gif\"><td align=\"left\"><table width=\"100%\" border=\"0\" cellpadding=\"5\">

<tr><td bgcolor=\"#ff9900\" colspan=\"3\">

Mapas</td></tr>");

//Imprimi toodos os dados achados na consulta SQL

\$cor = 0;

while (\$lista = mssql_fetch_array(\$servidor)){

\$ChaveServidor = \$lista['ChaveServidor'];

\$Titulo = \$lista['Titulo'];

\$Descricao = \$lista['Descricao'];

\$Capacidade = \$lista['Capacidade'];

\$NomeArquivoMapa = \$lista['NomeArquivoMapa'];

\$NomeArquivoTextura = \$lista['NomeArquivoTextura'];

\$LinkDownload = \$lista['LinkDownload'];

\$imagem = \$lista['ChaveServidor'];

if(strlen(\$imagem) < 6){

\$tamanho = 6 - strlen(\$imagem);

\$imagemjaux = \$imagem;

\$imagem = "";

for(\$x = 0; \$x < \$tamanho; \$x++){

\$imagem .= 0;

}

\$imagem .= \$imagemjaux;

\$imagem .= ".jpg";

```

    }

    if(($cor % 2) != 0){
        print("<tr>

                                <td width=\"110\"
                                bgcolor=\"\#eeeeee\" valign=\"top\"><a href=\"\$LinkDownload\"><img
                                src=\"http://www.colorgames.com.br/3dblah/Servers/$imagem\" width=\"100\"
                                height=\"100\" border=\"0\"></a></td>

                                <td bgcolor=\"\#eeeeee\"
                                valign=\"top\"><font color=\"\#000000\" size=\"4\" face=\"Verdana, Arial, Helvetica,
                                sans-serif\"><a href=\"\$LinkDownload\" style=\"color: \#000000\">
                                $Titulo</a></font><br><br><font color=\"\#000000\" size=\"2\" face=\"Verdana,
                                Arial, Helvetica, sans-serif\"><a href=\"index.php?pagina=noticia&id=$id\"
                                style=\"color: \#000000\">$Descricao</a><br><br>Capacidade:
                                $Capacidade<br>Tamanho:</font>

                                <td bgcolor=\"\#eeeeee\"
                                width=\"94\" valign=\"bottom\"><a href=\"\$LinkDownload\"><img
                                src=\"imagens/download.png\" border=\"0\"></a></td>

                                </tr>");
        }else{
            print("<tr>

                                <td width=\"110\" bgcolor=\"\#ffffff\"
                                valign=\"top\"><a href=\"\$LinkDownload\"><img
                                src=\"http://www.colorgames.com.br/3dblah/Servers/$imagem\" width=\"100\"
                                height=\"100\" border=\"0\"></a></td>

                                <td bgcolor=\"\#ffffff\"
                                valign=\"top\"><font color=\"\#000000\" size=\"4\" face=\"Verdana, Arial, Helvetica,
                                sans-serif\"><a href=\"index.php?pagina=noticia&id=$id\" style=\"color: \#000000\">
                                $Titulo</a></font><br><br><font color=\"\#000000\" size=\"2\" face=\"Verdana,
                                Arial, Helvetica, sans-serif\"><a href=\"index.php?pagina=noticia&id=$id\"

```

```

style="\color: #000000\">$Descricao</a><br><br>Capacidade:
$Capacidade<br>Tamanho:</font>

<td bgcolor="\#ffffff" width="\94"
valign="\bottom\ "><a href="\$LinkDownload\ "></a></td>

</tr>");

    }
    $cor++;
}

print("<tr><td bgcolor="\#ffffff" align="\right\ " colspan="\3\ "><font
size="\2\ " face="\Verdana, Arial, Helvetica, sans-serif\ "><a href="\download.php\ "
style="\color: 000000\ "><b>Voltar</b><a></font></tr></td>

</table></td> </tr>");

}

print("</table>
</td>

</tr>
</table>

<table width="\760\ " border="\0\ " cellspacing="\0\ " cellpadding="\0\ ">
<tr>
<td><object classid="\clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\ "
codebase="\http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\ " width="\760\ " height="\100\ ">
<param name="\movie\ " value="\rodape.swf\ ">
<param name="\quality\ " value="\high\ ">
<embed src="\rodape.swf\ " quality="\high\ "
pluginspage="\http://www.macromedia.com/go/getflashplayer\ "
type="\application/x-shockwave-flash\ " width="\760\ "
height="\100\ "></embed></object></td>

</tr>

```



```

        </table></td>
    </tr>
</table>
</body>
</html>");
}
?>

```

Modelo.php

```

<?php
//Verifica se o usuário está logado se não estiver redireciona para o login
session_start();

if (!isset($_SESSION['idblah'])){
    header("Location: index.php?pagina=adquira");
}
else{
    print("<html>
<head>
<title>| 3D BLAH |</title>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>
</head>

<body bottommargin='0' leftmargin='0' topmargin='0' rightmargin='0'>
<table width='100%' height='100%' border='0' cellpadding='0'
cellspacing='0'>
    <tr>
        <td align='center' valign='top' background='imagens/fundo.gif'>
            <table width='760' border='0' cellspacing='0' cellpadding='0'>
                <tr>

```

```

        <td height=\"25\">&nbsp;</td>
    </tr>
</table>

<table width=\"760\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">
    <tr>
        <td><object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\" width=\"760\" height=\"128\">
            <param name=\"movie\" value=\"topo.swf\">
            <param name=\"quality\" value=\"high\">
            <embed src=\"topo.swf\" quality=\"high\"
pluginspage=\"http://www.macromedia.com/go/getflashplayer\"
type=\"application/x-shockwave-flash\" width=\"760\"
height=\"128\"></embed></object></td>
    </tr>
</table>");
    if(isset($_SESSION['idblah'])){
        $nome = $_SESSION['nomeblah'];
        print("<table width=\"760\" border=\"0\" cellpadding=\"0\"
cellpadding=\"7\" background=\"imagens/corpo.gif\">
    <tr>
        <td height=\"15\" valign=\"top\">
            <table width=\"100%\" border=\"0\" cellpadding=\"2\">
                <tr>
                    <td bgcolor=\"#eeeeee\"
align=\"right\"><font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>$nome</b>, você est  logado. <a href=\"logout.php\" style=\"color:
#000000\"><b>Sair</b></a>&nbsp;&nbsp;&nbsp;</font></td>
                </tr>
            </table>
        </td>
    </tr>
</table>
    </td>

```

```

        </tr>
    </table>");
}
print("<table width=\"760\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">
<tr>
    <td height=\"270\" align=\"left\" valign=\"top\"
background=\"imagens/corpo.gif\">
        <table width=\"100%\" border=\"0\" cellpadding=\"0\"
cellpadding=\"15\">");

        //Chama a página conect.php
        require_once("conect.php");

        //Realizacunsulta SQL
        $modelo = mssql_query("SELECT * FROM Modelo");

        if($row_modelo = mssql_num_rows($modelo)){
            print("<tr background=\"imagens/corpo.gif\">
            <td>
                <font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><strong>Instru&ccedil;&otilde;es:</strong> Ap&ocirc;e; baixar o
                arquivo em seu pc extraia-o na pasta onde o 3D BLAH est&aacute;
                instalado.</font>
            </td>
            </tr>
            <tr background=\"imagens/corpo.gif\"><td align=\"left\"><table
width=\"100%\" border=\"0\" cellpadding=\"5\">
                <tr><td bgcolor=\"#ff9900\" colspan=\"2\">
                    <font color=\"#ffffff\" size=\"2\" face=\"Verdana, Arial, Helvetica,
sans-serif\"><b>Modelos</b></font></tr></td>");

```



```

        $cor++;
    }
    print("<tr><td bgcolor=\"#ffffff\" align=\"right\" colspan=\"2\"><font
size=\"2\" face=\"Verdana, Arial, Helvetica, sans-serif\"><a href=\"download.php\"
style=\"color: 000000\"><b>Voltar</b><a></font></tr></td>
</table></td> </tr>");
}

print("</table>
</td>

</tr>
</table>
<table width=\"760\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">
<tr>
<td><object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,29,0\" width=\"760\" height=\"100\">
<param name=\"movie\" value=\"rodape.swf\">
<param name=\"quality\" value=\"high\">
<embed src=\"rodape.swf\" quality=\"high\"
pluginspage=\"http://www.macromedia.com/go/getflashplayer\"
type=\"application/x-shockwave-flash\" width=\"760\"
height=\"100\"></embed></object></td>
</tr>
</table></td>
</tr>
</table>
</body>
</html>");
}
?>

```

Logout.php

```

<?php
//Inicia a seção
session_start();

//Entra com valor nulo nas seções
$_SESSION['idblah'] = "";
$_SESSION['nomeblah'] = "";

//Destroi as variaveis
unset($_SESSION['idblah']);
unset($_SESSION['nomeblah']);

//REDIRECIONA PARA A TELA DE LOGIN
header("Location: index.php");
?>

```

Erro404.php

```

<table width="760" border="0" cellspacing="20" cellpadding="0">
  <tr>
    <td height="270" align="center" valign="middle"> <table width="500" border="0"
cellspacing="0" cellpadding="0">
      <tr>
        <td width="127"></td>
        <td width="10" bgcolor="#FF9900"> </td>

```

```

        <td align="center" bgcolor="#eeeeee"><p><font color="#000000" size="2"
face="Verdana, Arial, Helvetica, sans-serif"><strong>N&atilde;o
        conseguimos encotrar a p&aacute;gina solicitada</strong></font></p>
        <p><strong><font color="#000000" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><a href="index.php" style="color: #000000">P&aacute;gina
        inicial</a></font></strong></p></td>
        <td width="10" bgcolor="#FF9900"> </td>
    </tr>
</table></td>
</tr>
</table>

```